

Prioritised Default Logic as Argumentation with Partial Order Default Priorities

Anthony P. Young, Sanjay Modgil, Odinaldo Rodrigues

Department of Informatics, King's College London, Strand, London, U.K.
{peter.young,sanjay.modgil,odinaldo.rodrigues}@kcl.ac.uk

20th September 2016

Abstract

We express Brewka's prioritised default logic (PDL) as argumentation using ASPIC⁺. By representing PDL as argumentation and designing an argument preference relation that takes the argument structure into account, we prove that the conclusions of the justified arguments correspond to the PDL extensions. We will first assume that the default priority is total, and then generalise to the case where it is a partial order. This provides a characterisation of non-monotonic inference in PDL as an exchange of argument and counter-argument, providing a basis for distributed non-monotonic reasoning in the form of dialogue.¹

Contents

1	Introduction	2
2	Background	4
2.1	Notation Used in this Paper	4
2.2	The ASPIC ⁺ Framework	4
2.3	Brewka's Prioritised Default Logic	8
3	From ASPIC⁺ to PDL	9
3.1	Representing PDL in ASPIC ⁺	9
3.2	A Suitable Argument Preference Relation	10
3.3	The Representation Theorem	18
3.3.1	Non-Blocked Defaults	18
3.3.2	Existence and Uniqueness of Stable Extensions	21
3.3.3	The Representation Theorem: Statement and Proof	24
3.4	Satisfaction of Rationality Postulates	27
3.4.1	The Stable Extension is Grounded	27

¹The results of Section 3 first appeared in the preprint [23] and have been published in the conference proceedings of AAMAS2016 [24]. This paper gives the full proofs of these results.

3.4.2	The Trivialisation and Rationality Theorems	31
3.4.3	Inconsistent Arguments	32
3.5	Summary	32
4	On Lifting the Assumption of a Total Order Default Priority	33
4.1	The Argument Preference Relation based on Partial Order Default Priorities	33
4.1.1	A Representation of Rules and their Ordering using Strings	33
4.1.2	Algorithm and Example Calculation	34
4.1.3	Properties of the Generalised SP Order	37
4.1.4	The Generalised Argument Preference Relation for $<_D$ Partial	39
4.2	The Representation Theorem for Partial Order Default Priorities	41
4.2.1	Linearisation of the Argument Preference Relation and Spanning Subgraphs	41
4.2.2	Existence of Stable Extensions in the Partial Order Case	42
4.2.3	Proof of the Representation Theorem	43
4.3	Satisfaction of Rationality Postulates	46
4.4	Summary	47
5	Conclusions	47
5.1	Related Work	47
5.2	Future Work	48

1 Introduction

Dung’s *abstract argumentation theory* [11] has become established as a means for unifying various nonmonotonic logics (NMLs) [4,20,22], where the inferences of a given NML can be interpreted as conclusions of justified arguments. Abstract argumentation defines “justified arguments” by making use of principles familiar in everyday reasoning and debate. This renders the process of inference in the NML transparent and amenable to human inspection and participation, and serves as a basis for distributed reasoning and dialogue.

More precisely, relating NMLs and argumentation is to endow the NML with *argumentation semantics*. This has already been done for default logic [11], logic programming [11], defeasible logic [12] and preferred subtheories [16]. This allows the application of argument game proof theories [15] to the process of inference in these NMLs, and the generalisation of these dialectical proof theories to distributed reasoning amongst computational agents, where agents can engage in argumentation-based dialogues [1,14,17].

Abstract argumentation has been upgraded to *structured argumentation theory* [3], one example of which is the *ASPIC⁺ framework for structured argumentation* [16]. In ASPIC⁺, arguments are constructed from premises and deductive or defeasible rules of inference. The conclusions of arguments can contradict each other and hence arguments can attack each other. A preference relation

over the arguments can be used to determine which attacks succeed as defeats. The arguments and defeats instantiate an abstract argumentation framework, where the justified arguments are determined using Dung’s method. The conclusions of the justified arguments are then identified with the nonmonotonic inferences from the underlying premises and rules of inference. The advantages of ASPIC⁺ are that the framework provides a systematic and general method of endowing non-monotonic logics with argumentation semantics, and identifies sufficient conditions on the underlying logic and preference relations that guarantee the satisfaction of various normatively rational desiderata [10].

This paper endows Brewka’s *prioritised default logic* (PDL) [7] with argumentation semantics. PDL is an important NML because it upgrades default logic (DL) [19] with an explicit priority relation over defaults, so that, for example, one can account for recent information taking priority over information in the distant past. PDL has also been used to represent the (possibly conflicting) beliefs, obligations, intentions and desires (BOID) of agents, and model how these different categories of mental attitudes override each other in order to generate goals and actions that attain those goals [9].

We prove a correspondence between inferences in PDL and the conclusions of the justified arguments defined by the argumentation semantics. We realise these contributions by appropriately representing PDL in ASPIC⁺. The main challenges involve understanding how priorities over defaults in PDL can be represented as an ASPIC⁺ argument preference relation, and then applying the properties of this preference relation to prove that the extensions of PDL correspond to the conclusions of justified arguments.

This paper has five sections. In Section 2, we review ASPIC⁺, abstract argumentation, and PDL. In Section 3 we present an instantiation of ASPIC⁺ to PDL when the default priority is total. The key results are the design of an appropriate argument preference relation (Section 3.2), and showing that this argument preference relation guarantees that the conclusions of the justified arguments correspond exactly to the PDL extensions by the *representation theorem* (Section 3.3). We then investigate some properties and directly prove that the normative rationality postulates of [10] are satisfied (Section 3.4).²

In Section 4 we lift the assumption that the priority order on the defaults is total. Following the pattern of the previous section we generalise the argument preference (Section 4.1) to accommodate for partial order default priorities. We then prove a generalised representation theorem (Sections 4.2) and prove a partial result concerning the satisfaction of the rationality postulates of [10] (Section 4.3). We conclude in Section 5 with suggestions for future work.

²But in this case we are not leveraging the properties of ASPIC⁺ to achieve this. We will discuss this point in Section 5.

2 Background

2.1 Notation Used in this Paper

In this paper: “:=” is read “is defined as”. WLOG stands for “without loss of generality”. \mathbb{N} denotes the set of natural numbers. We denote set difference with $-$. For two sets A, B , $A \ominus B := (A - B) \cup (B - A)$ denotes their symmetric difference. If $f : X \rightarrow Y$ is a function and $A \subseteq X$, $f(A) \subseteq Y$ is the image set of A in Y under f . For a set X its power set is $\mathcal{P}(X)$ and its finite power set (set of all finite subsets) is $\mathcal{P}_{\text{fin}}(X)$. $X \subseteq_{\text{fin}} Y$ iff X is a finite subset of Y , therefore $X \in \mathcal{P}_{\text{fin}}(Y) \Leftrightarrow X \subseteq_{\text{fin}} Y$. Undefined quantities are denoted by $*$, for example $1/0 = *$ in the real numbers. Order isomorphism is denoted by \cong .

If $\langle P, \lesssim \rangle$ is a preordered set then the strict version of the preorder is $a < b \Leftrightarrow [a \lesssim b, b \not\lesssim a]$, which is also a strict partial order. If $<$ is a strict partial order on P and $U \subseteq P$, then we define the set $\max_{<} U := \{x \in U \mid (\forall y \in U) x \not< y\} \subseteq U$, i.e. the set of all $<$ -maximal elements of U . We define the set $\min_{<} U$ analogously. For a set X we define the set of possible strict partial orders on X to be $PO(X) := \{< \subseteq X^2 \mid < \text{ is a strict partial order}\}$. Similarly, the set of all possible strict total orders on X is $TO(X) := \{< \subseteq X^2 \mid < \text{ is a strict total order}\} \subset PO(X)$. We will use the terms “total (order)” and “linear (order)” interchangeably. We will also call totally ordered sets either “tosets” or “chains”.

2.2 The ASPIC⁺ Framework

Abstract argumentation abstracts from the internal logical structure of arguments, the nature of defeats and how they are determined by preferences, and consideration of the conclusions of the arguments [11]. However, these features are referenced when studying whether any given logical instantiation of a framework yields complete extensions that satisfy the rationality postulates of [10]. ASPIC⁺ [16] provides a structured account of abstract argumentation, allowing one to reference the above features, while at the same time accommodating a wide range of instantiating logics and preference relations in a principled manner. ASPIC⁺ then identifies conditions under which complete extensions defined by the arguments, attacks and preferences, satisfy the rationality postulates of [10].

In ASPIC⁺, the tuple $\langle \mathcal{L}, -, \mathcal{R}_s, \mathcal{R}_d, n \rangle$ is an *argumentation system*, where \mathcal{L} is a logical language and $- : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{L})$ is the *contrary function* $\theta \mapsto \bar{\theta}$ where $\bar{\theta}$ is the set of wffs that are inconsistent with θ . Let $\theta_1, \dots, \theta_m, \phi \in \mathcal{L}$ be wffs for $m \in \mathbb{N}$, \mathcal{R}_s is the *set of strict inference rules* of the form $(\theta_1, \dots, \theta_m \rightarrow \phi)$, denoting that if $\theta_1, \dots, \theta_m$ are true then ϕ is also true, and \mathcal{R}_d is the *set of defeasible inference rules* of the form $(\theta_1, \dots, \theta_m \Rightarrow \phi)$, denoting that if $\theta_1, \dots, \theta_m$ are true then ϕ is tentatively true. Note $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$. For a strict or defeasible rule $r = (\theta_1, \dots, \theta_m \rightarrow / \Rightarrow \phi)$, we define $\text{Ante}(r) := \{\theta_1, \dots, \theta_m\} \subseteq_{\text{fin}} \mathcal{L}$,³ and $\text{Cons}(r) := \phi \in \mathcal{L}$. Finally, $n : \mathcal{R}_d \rightarrow \mathcal{L}$ is a *partial* function that assigns a *name* to *some* of the defeasible rules. For any $S \subseteq \mathcal{L}$ we define the set $\text{Cl}_{\mathcal{R}_s}(S) \subseteq \mathcal{L}$ to be the smallest superset of S that also contains $\text{Cons}(r)$ for

³Note it is possible to have $m = 0$ and hence $\text{Ante}(r) = \emptyset$.

all $r \in \mathcal{R}_s$ such that $\text{Ante}(r) \subseteq \text{Cl}_{\mathcal{R}_s}(S)$. We call $\text{Cl}_{\mathcal{R}_s}$ the *closure under strict rules operator*.

In ASPIC^+ , a *knowledge base* is a set $\mathcal{K} := \mathcal{K}_n \cup \mathcal{K}_p \subseteq \mathcal{L}$ where \mathcal{K}_n is the set of *axioms* and \mathcal{K}_p is the set of *ordinary premises*. Note that $\mathcal{K}_n \cap \mathcal{K}_p = \emptyset$. Given an argumentation system and \mathcal{K} , *arguments* are defined inductively:

1. (Base) $[\theta]$ is a *singleton argument* with $\theta \in \mathcal{K}$, conclusion $\text{Conc}([\theta]) := \theta$, premise set $\text{Prem}([\theta]) := \{\theta\} \subseteq \mathcal{K}$, top rule $\text{TopRule}([\theta]) := *$ and set of subarguments $\text{Sub}([\theta]) := \{\emptyset\}$.
2. (Inductive) Let A_1, \dots, A_n be arguments with respective conclusions $\text{Conc}(A_1), \dots, \text{Conc}(A_n)$ and premise sets $\text{Prem}(A_1), \dots, \text{Prem}(A_n)$. If there is a rule $r := (\text{Conc}(A_1), \dots, \text{Conc}(A_n) \rightarrow / \Rightarrow \phi) \in \mathcal{R}$, then $B := [A_1, \dots, A_n \rightarrow / \Rightarrow \phi]$ is also an argument with $\text{Conc}(B) = \phi$, premises $\text{Prem}(B) := \bigcup_{i=1}^n \text{Prem}(A_i)$, $\text{TopRule}(B) = r \in \mathcal{R}$ and set of subarguments $\text{Sub}(B) := \{B\} \cup \bigcup_{i=1}^n \text{Sub}(A_i)$.

Let \mathcal{A} be the (unique) set of all arguments freely constructed following the above rules. It is clear that arguments are finite objects in that each argument has finitely many premises, and take finitely many rules to reach its conclusion. We define the *conclusion map* $\text{Conc} : \mathcal{A} \rightarrow \mathcal{L} : A \mapsto \text{Conc}(A)$. We can generalise this to arbitrary *sets* of arguments (abuse of notation):

$$\text{Conc} : \mathcal{P}(\mathcal{A}) \rightarrow \mathcal{P}(\mathcal{L}) : S \mapsto \text{Conc}(S) := \bigcup_{A \in S} \text{Conc}(A). \quad (2.1)$$

Two strict or defeasible rules are *equal* iff they have the same antecedent sets, consequents and name syntactically in the underlying \mathcal{L} . Two arguments are *equal* iff they are constructed identically as described above. More precisely, we can define equality of arguments inductively. The base case would be two singleton arguments $[\theta]$, $[\phi]$ are equal iff θ and ϕ are *syntactically* the same formulae. Given n arguments A_1, \dots, A_n and two equal rules r_1 and r_2 (either both strict or both defeasible) with antecedent $\{\text{Conc}(A_i)\}_{i=1}^n$, such that B_1 is the rule r_1 appended to the A_i 's, and B_2 is the rule r_2 appended to the A_i 's, then B_1 and B_2 are equal arguments.

We say A is a *subargument* of B iff $A \in \text{Sub}(B)$ and we write $A \subseteq_{\text{arg}} B$. We say A is a *proper subargument* of B iff $A \in \text{Sub}(B) - \{B\}$ and we write $A \subset_{\text{arg}} B$. It can be shown that \subseteq_{arg} is a preorder on $\text{Sub}(B)$. A set of arguments is *subargument closed* iff it is \subseteq_{arg} -down closed. Clearly, for every defeasible rule r in an argument A , there is a subargument of A with r as its top rule, by the inductive construction of arguments.

An argument $A \in \mathcal{A}$ is *firm* iff $\text{Prem}(A) \subseteq \mathcal{K}_n$. Further, $\text{SR}(A) \subseteq \mathcal{R}_s$ is the set of strict rules applied in constructing A , and $\text{DR}(A) \subseteq \mathcal{R}_d$ is the set of defeasible rules applied in constructing A . We also define $\text{Prem}_n(A) := \text{Prem}(A) \cap \mathcal{K}_n$ and $\text{Prem}_p(A) := \text{Prem}(A) \cap \mathcal{K}_p$. An argument A is *strict* iff $\text{DR}(A) = \emptyset$, else A is *defeasible*. We can generalise $\text{DR}(\cdot)$ to sets as well just like Equation 2.1 for $\text{Conc}(\cdot)$.

Given $R \subseteq \mathcal{R}_d$, we introduce *the set of all arguments freely constructed with defeasible rules restricted to those in R* as the set $Args(R) \subseteq \mathcal{A}$, which are all arguments with premises in \mathcal{K} , strict rules in \mathcal{R}_s and defeasible rules in R . Formally, $Args(R)$ is defined inductively just as how arguments are constructed, but with the choice of defeasible rules restricted to those in R . It is easy to show that this definition is equivalent to

$$A \in Args(R) \Leftrightarrow DR(A) \subseteq R. \quad (2.2)$$

Clearly, $Args(\mathcal{R}_d) = \mathcal{A}$. Given R , $Args(R)$ exists and is unique.

Let $S \subseteq_{\text{fin}} \mathcal{A}$. The *set of all strict extensions of S* is the set $StExt(S)$ where

$$\begin{aligned} A \in StExt(S) \Leftrightarrow & DR(A) = DR(S), SR(A) \supseteq SR(S), \\ & Prem_p(A) = Prem_p(S), Prem_n(A) \supseteq Prem_n(S). \end{aligned}$$

A set $S \subseteq \mathcal{A}$ is *closed under strict extensions* iff for all $T \subseteq_{\text{fin}} S$, $StExt(T) \subseteq S$.

Lemma 2.1. *The set $Args(R)$, for any $R \subseteq \mathcal{R}_d$, is closed under strict extensions and subarguments.*

Proof. If $A \in Args(R)$ and $B \subseteq A$, then $DR(B) \subseteq DR(A) \subseteq R$ so $DR(B) \subseteq R$ and hence $B \in Args(R)$, therefore $Args(R)$ is subargument closed. Now let $T \subseteq_{\text{fin}} Args(R)$, so for all $B \in T$, $DR(B) \subseteq R$, therefore $DR(T) := \bigcup_{B \in T} DR(B) \subseteq R$. Let $A \in StExt(T)$, then $DR(A) = DR(T) \subseteq R$ and hence $A \in Args(R)$. Therefore, $StExt(T) \subseteq Args(R)$, therefore $Args(R)$ is closed under strict extensions. \square

An argument A *attacks* another argument B , denoted as $A \rightarrow B$, iff at least one of the following hold, where:

1. A is said to *undermine* attack B on the (singleton) subargument $B' = [\phi]$ iff there is some $\phi \in Prem_p(B)$ such that $Conc(A) \in \overline{\phi}$.
2. A is said to *rebut* attack B on the subargument B' iff there is some $B' \subseteq_{\text{arg}} B$ such that $r := TopRule(B') \in \mathcal{R}_d$, $\phi := Cons(r)$ and $Conc(A) \in \overline{\phi}$.
3. A is said to *undercut* attack B on the subargument B' iff there is some $B' \subseteq_{\text{arg}} B$ such that $r := TopRule(B') \in \mathcal{R}_d$ and $Conc(A) \in \overline{n(r)}$.

See [16, Section 2] for a further discussion of why attacks are distinguished in this way. We abuse notation to define the *attack relation* as $\rightarrow \subseteq \mathcal{A}^2$ such that $(A, B) \in \rightarrow \Leftrightarrow A \rightarrow B$. Notice that by the transitivity of \subseteq_{arg} , if $A \rightarrow B$ and $B \subseteq_{\text{arg}} C$, then $A \rightarrow C$.

A preference relation over arguments is then used to determine which attacks succeed as defeats. We denote the preference $\preceq \subseteq \mathcal{A}^2$ (not necessarily a preorder for now) such that $A \preceq B \Leftrightarrow B$ is at least as preferred as A . Strict preference and equivalence are, respectively, $A \prec B \Leftrightarrow [A \preceq B, B \not\preceq A]$ and $A \approx B \Leftrightarrow [A \preceq B, B \preceq A]$. We define a *defeat* as

$$A \hookrightarrow B \Leftrightarrow (\exists B' \subseteq_{\text{arg}} B) [A \rightarrow B', A \not\prec B']. \quad (2.3)$$

That is to say, A defeats B (on B') iff A attacks B on the subargument B' , and B' is not strictly preferred to A . Notice the comparison is made at the subargument B' instead of the whole argument B . We abuse notation to define the *defeat relation* as $\hookrightarrow \subseteq \mathcal{A}^2$ such that $(A, B) \in \hookrightarrow \Leftrightarrow A \hookrightarrow B$. A set of arguments $S \subseteq \mathcal{A}$ is *conflict-free* (cf) iff $\hookrightarrow \cap S^2 = \emptyset$.⁴ Notice that by the transitivity of \subseteq_{arg} and that the preference comparison is made at the defeated subargument, if $A \hookrightarrow B$ and $B \subseteq_{\text{arg}} C$, then $A \hookrightarrow C$. As relations, $\hookrightarrow \subseteq \rightarrow$.

Preferences between arguments are calculated from the argument structure by comparing arguments at their fallible components, i.e. the ordinary premises \mathcal{K}_p and defeasible rules \mathcal{R}_d with preorders \lesssim_K and \lesssim_D respectively, where (e.g.) $r_1 \lesssim_D r_2$ iff r_2 is *just as preferred or more preferred* than r_1 (and analogously for \lesssim_K). These preorders are then aggregated to a set-comparison relation \sqsubseteq between the sets of premises and / or defeasible rules of the arguments, and then finally to $\lesssim \subseteq \mathcal{A}^2$, following the method in [16, Section 5].⁵ We will use a modified version of this lifting, which will be explained in Section 3.2.

Given the preference relation \lesssim between arguments, we call the structure $\langle \mathcal{A}, \rightarrow, \lesssim \rangle$ an *ASPIC⁺ SAF* (structured argumentation framework), or *attack graph*. Its corresponding *defeat graph* is $\langle \mathcal{A}, \hookrightarrow \rangle$, where \hookrightarrow is defined in terms of \rightarrow and \lesssim as in Equation 2.3.

Given $\langle \mathcal{A}, \hookrightarrow \rangle$ one can then evaluate the extensions under Dung's abstract argumentation semantics, and thus identify the inferences defined by argumentation as the conclusions of the justified arguments. We now recap the key definitions of [11]. An *argumentation framework* is a directed graph $\langle \mathcal{A}, \hookrightarrow \rangle$, where \mathcal{A} is the set of arguments and $\hookrightarrow \subseteq \mathcal{A}^2$ is the *defeat relation*, such that $A \hookrightarrow B$ means A is a (successful) counterargument against B . The argumentation frameworks we consider are defeat graphs, but this is a general definition.

Let $S \subseteq \mathcal{A}$ and $A, B \in \mathcal{A}$. S *defeats* B iff $(\exists A \in S) A \hookrightarrow B$. S is *conflict-free* (cf) iff $\hookrightarrow \cap S^2 = \emptyset$. S *defends* A iff $(\forall B \in \mathcal{A}) [B \hookrightarrow A \Rightarrow S \text{ defeats } B]$. The *characteristic function* is $\chi : \mathcal{P}(\mathcal{A}) \rightarrow \mathcal{P}(\mathcal{A})$, such that $\chi(S) := \{A \in \mathcal{A} \mid S \text{ defends } A\} \subseteq \mathcal{A}$. S is an *admissible extension* iff S is cf and $S \subseteq \chi(S)$. An admissible extension S is: a *complete extension* iff $S = \chi(S)$; a *preferred extension* iff S is a \subseteq -maximal complete extension; the *grounded extension* iff S is the \subseteq -least complete extension; a *stable extension* iff S is complete and defeats all arguments $B \in \mathcal{A} - S$.

Let $\mathcal{S} := \{\text{complete, preferred, grounded, stable}\}$ be the set of *Dung semantics*. An argument $A \in \mathcal{A}$ is *sceptically (credulously) justified* under the semantics $s \in \mathcal{S}$ iff A belongs to all (at least one) of the s -extensions of $\langle \mathcal{A}, \hookrightarrow \rangle$.

Instantiations of ASPIC⁺ should satisfy some properties to ensure they are rational [10]. Given an instantiation let $\langle \mathcal{A}, \rightarrow, \lesssim \rangle$ be its ASPIC⁺ attack graph

⁴Note that [16] studies two different notions of cf sets: one where no two arguments *attack* each other, and the other where no two arguments *defeat* each other. We choose the latter notion of cf as this is more commonplace in argumentation formalisms that distinguish between attacks and defeats, e.g. in [18].

⁵Note there are many other ways to lift a preference $<$ on a set of objects X to compare subsets of X in various ways that are “compatible” with $<$ [2].

with corresponding defeat graph $\langle \mathcal{A}, \hookrightarrow \rangle$. Let \mathcal{E} be any complete extension. The *Caminada-Amgoud rationality postulates* state:

1. (Subargument closure) \mathcal{E} is subargument closed.
2. (Closure under strict rules) \mathcal{E} satisfies $\text{Conc}(\mathcal{E}) = \text{Cl}_{\mathcal{R}_s}[\text{Conc}(\mathcal{E})]$, where $\text{Conc}(\mathcal{E})$ is defined in Equation 2.1.
3. (Consistency) $\text{Conc}(\mathcal{E})$ is consistent.⁶

An ASPIC⁺ instantiation is *normatively rational* iff it satisfies these rationality postulates. These postulates may be proved directly given an instantiation. ASPIC⁺ also identifies sufficient conditions for an instantiation to satisfy these postulates [16, Section 4], which we will discuss in Section 5.

2.3 Brewka’s Prioritised Default Logic

In this section we recap Brewka’s prioritised default logic (PDL) [7]. We work in first order logic (FOL) of arbitrary signature where the set of first-order formulae is \mathcal{FL} and the set of closed first order formulae⁷ a.k.a. *sentences* is $\mathcal{SL} \subseteq \mathcal{FL}$, with the usual quantifiers and connectives. Entailment is denoted by \models . Logical equivalence of formulae is denoted by \equiv . Given $S \subseteq \mathcal{FL}$, the *deductive closure* of S is $\text{Th}(S)$, and given $\theta \in \mathcal{FL}$, the *addition operator* $+: \mathcal{P}(\mathcal{FL}) \times \mathcal{FL} \rightarrow \mathcal{P}(\mathcal{FL})$ is defined as $S + \theta := \text{Th}(S \cup \{\theta\})$.

A *normal default* is an expression $\frac{\theta:\phi}{\phi}$ where $\theta, \phi \in \mathcal{FL}$ and read “if θ is the case and ϕ is consistent with what we know, then jump to the conclusion ϕ even if it does not deductively follow”. In this case we call θ the *antecedent* and ϕ the *consequent*. A normal default $\frac{\theta:\phi}{\phi}$ is *closed* iff $\theta, \phi \in \mathcal{SL}$. We will assume all defaults are closed and normal unless stated otherwise. Given $S \subseteq \mathcal{SL}$, a default is *active (in S)* iff $[\theta \in S, \phi \notin S, \neg\phi \notin S]$.

A *finite prioritised default theory* (PDT) is a structure $T := \langle D, W, < \rangle$, where *the set of facts* $W \subseteq \mathcal{SL}$ is not necessarily finite and $\langle D, < \rangle$ is a *finite* strict partially ordered set of defaults that nonmonotonically extend W . The priority relation is such that $d' < d \Leftrightarrow d$ is *more*⁸ *prioritised* than d' . All PDTs in this paper are finite.

The inferences of a PDT $T = \langle D, W, < \rangle$ are defined by its extensions. Let $<^+ \supseteq <$ be a linearisation of $<$. A *prioritised default extension (with respect to $<^+$)* (PDE) is a set $E := \bigcup_{i \in \mathbb{N}} E_i \subseteq \mathcal{SL}$ built inductively as:

⁶Notice by properties 2 and 3 above $\text{Cl}_{\mathcal{R}_s}(\text{Conc}(\mathcal{E}))$ is consistent. ASPIC⁺ distinguishes this into direct and indirect consistency given that \mathcal{R}_s is in general arbitrary and do not have to be the rules of inference of classical logic. We will not make this distinction because our underlying logic will be first order logic (FOL) (Section 3.1). Further, consistency in the abstract logic of ASPIC⁺ is expressed in terms of the contrary function, but since our contrary function will just be classical negation, we can take the usual meaning of consistency in FOL.

⁷i.e. first order formulae without free variables

⁸We have defined the order dually to [7] so as to comply with orderings over the ASPIC⁺ defeasible inference rules. This goes *against* the tradition in NML where the *smaller* item in $<$ is the *more* preferred one.

$$E_0 := Th(W) \text{ and} \quad (2.4)$$

$$E_{i+1} := \begin{cases} E_i + \phi, & \text{if property 1} \\ E_i, & \text{else} \end{cases} \quad (2.5)$$

where “property 1” abbreviates “ ϕ is the consequent of the $<^+$ -*greatest*⁹ default d active in E_i ”. Intuitively, one first generates all classical consequences from the facts W , and then iteratively adds the nonmonotonic consequences from the highest priority default to the lowest. Notice if W is inconsistent then $E_0 = E = \mathcal{FL}$. For this paper we will assume W is always consistent.

For finite D it can be shown that the ascending chain $E_i \subseteq E_{i+1}$ stabilises at some finite $i \in \mathbb{N}$ and that E is consistent provided that W is consistent. E does not have to be unique because there are many distinct linearisations of $<$. We say T *sceptically infers* $\theta \in \mathcal{SL}$ iff $\theta \in E$ for *all* extensions E of T .

A PDT T for which $<$ is a strict total order is a *linearised PDT* (LPDT). If $<$ is total then there is only one way to apply the defaults in D by Equation 2.5, hence the extension is unique. We will use the notation $<^+$ to emphasise that the priority is total, and the notation T^+ to denote an arbitrary LPDT.

For the rest of this paper, if we declare T to be a PDT, we mean $T = \langle D, W, < \rangle$ where each component is defined above, and we make no further assumptions on each component. If we declare T^+ to be an LPDT, we mean $T^+ = \langle D, W, <^+ \rangle$ where $<^+$ is a strict total order on D .

3 From ASPIC⁺ to PDL

3.1 Representing PDL in ASPIC⁺

We now instantiate ASPIC⁺ to PDL. Let $T^+ := \langle D, W, <^+ \rangle$ be an LPDT.¹⁰

1. Our arguments are expressed in FOL, so our set of wffs \mathcal{L} is \mathcal{FL} .
2. The contrary function $- : \mathcal{FL} \rightarrow \mathcal{P}(\mathcal{FL})$ syntactically defines conflict in terms of classical negation. For all $\theta \in \mathcal{FL}$, $\bar{\theta} = \{-\theta\}$ unless θ has the syntactic form $\neg\phi$ for some $\phi \in \mathcal{FL}$, then $\bar{\theta} = \{\phi\}$. As $\bar{\theta}$ is singleton, we will abuse notation and write $\bar{\theta}$ to refer to its element.
3. The set of strict rules \mathcal{R}_s characterises inference in FOL. Notice \mathcal{R}_s is *closed under transposition*, i.e. for all $1 \leq i \leq n \in \mathbb{N}^+$,

$$\begin{aligned} &(\theta_1, \theta_2, \dots, \theta_{i-1}, \theta_i, \theta_{n+1}, \dots, \theta_n \rightarrow \phi) \in \mathcal{R}_s \\ \Rightarrow &(\theta_1, \theta_2, \dots, \theta_{i-1}, -\phi, \theta_{n+1}, \dots, \theta_n \rightarrow -\theta_i) \in \mathcal{R}_s. \end{aligned}$$

We leave the proof theory implicit. $Cl_{\mathcal{R}_s}$ instantiates to deductive closure.

⁹See Footnote 8.

¹⁰We will lift this assumption of a total order priority in Section 4.

4. The set of defeasible rules \mathcal{R}_d is defined as:

$$\mathcal{R}_d := \left\{ (\theta \Rightarrow \phi) \left| \frac{\theta : \phi}{\phi} \in D \right. \right\},$$

with the naming function $n \equiv *$. Clearly, there is a bijection f where

$$f : D \rightarrow \mathcal{R}_d : \frac{\theta : \phi}{\phi} \mapsto f\left(\frac{\theta : \phi}{\phi}\right) := (\theta \Rightarrow \phi) \quad (3.1)$$

and we will define the *strict version of the* preorder \leq_D over \mathcal{R}_d as¹¹

$$(\theta \Rightarrow \phi) <_D (\theta' \Rightarrow \phi') \Leftrightarrow \frac{\theta : \phi}{\phi} <^+ \frac{\theta' : \phi'}{\phi'}. \quad (3.2)$$

We can see that the strict toset $\langle \mathcal{R}_d, <_D \rangle$ is order isomorphic to $\langle D, <^+ \rangle$.

5. The set of axiom premises is $\mathcal{K}_n = W$, because we take W to be the set of facts. Furthermore, $\mathcal{K}_p = \emptyset$.

The set \mathcal{A} of ASPIC⁺ arguments are defined as in Section 2.2.¹² All arguments are firm because $\mathcal{K}_p = \emptyset$, and so there are no undermining attacks. As n is undefined, no attack can be an undercut. Therefore, we only have rebut attacks,

$$A \rightarrow B \Leftrightarrow (\exists B', B'' \subseteq_{\text{arg}} B) \ B' = [B'' \Rightarrow \overline{\text{Conc}(A)}]. \quad (3.3)$$

Defeats are defined as in Equation 2.3. In the next section, we will define the argument preference \succsim , based on the strict total order $<_D$ over \mathcal{R}_d .

3.2 A Suitable Argument Preference Relation

We wish to define a suitable argument preference relation such that the conclusion set of the stable extension defined by \hookrightarrow corresponds to the extension of the underlying PDT.¹³ The first place to look for such a relation is in the existing relations of ASPIC⁺ [16, Definition 19]. However, simple counterexamples can be devised to show the inferences of the PDT and its argumentation counterpart do not correspond.

The difference between PDL and ASPIC⁺ is in how blocked defaults are treated. In PDL, blocked defaults are simply excluded from the extension. In ASPIC⁺, it is possible to construct arguments with defeasible rules that correspond to blocked defaults. If $<^+ \cong <_D$ such that $<^+$ is arbitrary, there is no guarantee that the blocked defaults will be positioned in the chain $<^+$ such that arguments with blocked defaults are always defeated by arguments with

¹¹From Footnote 8, we do not need to define $<_D$ as the order-theoretic dual to $<^+$, avoiding potential confusion as to which item is more preferred.

¹²As \mathcal{R}_s is a countably infinite set, \mathcal{A} is also a countably infinite set.

¹³In Section 3.4, we will show that for the resulting defeat graphs there is only one extension in that is stable, grounded and preferred.

only non-blocked defaults.¹⁴ To ensure that arguments with blocked defaults are defeated and hence the conclusions of the justified arguments form the extension of the PDT, we need to rearrange the rules in \mathcal{R}_d to take into account the structure of arguments. ASPIC⁺ does allow for explicit reference to argument structure, i.e. we can tell which defeasible rules precede which within an argument.

Rearranging $<_D$ to take argument structure into account captures the PDL meaning of “active” default, because defaults are added to E_i when its prerequisite is inferred. This rearrangement will mean that every defeasible rule r corresponding to a blocked default will be less preferred than the rules which make up arguments that rebut the argument with r as its top rule. We now devise a new ASPIC⁺ argument preference relation which incorporates the argument structure into the preorder $<_D$.

More formally, given any strict total order $<_D$ on \mathcal{R}_d , we first define a transformation $<_D \mapsto <_{SP}$, where the subscript SP stands for *structure-preference*. This sorts the defeasible rules in a way compatible with both the priority $<_D$ and their logical structure.

The set \mathcal{R}_d is finite because we have assumed that D is finite (Sections 2.3 and 3.1). Let $1 \leq i \leq |\mathcal{R}_d| =: N \in \mathbb{N}$. We define $a_i \in \mathcal{R}_d$ to be the $<_D$ -greatest element of the following set:

$$\left\{ r \in \mathcal{R}_d \mid \text{Ante}(r) \subseteq \text{Conc} \left[\text{Args} \left(\bigcup_{k=1}^{i-1} \{a_k\} \right) \right] \right\} - \bigcup_{j=1}^{i-1} \{a_j\}. \quad (3.4)$$

The intuition is: a_1 is the most preferred rule whose antecedent is inferred by the conclusions of all strict arguments, a_2 is the next most preferred rule, whose antecedent is amongst the conclusions of all arguments having *at most* a_1 as a defeasible rule. Similarly, a_3 is the next most preferred rule, whose antecedent is amongst the conclusions of all arguments having *at most* a_1 and a_2 as defeasible rules, and so on until all of the rules of \mathcal{R}_d are exhausted. Notice that the second union after the set difference in Equation 3.4 ensures that once a rule is applied it cannot be applied again. We then define $<_{SP}$ as (notice the dual order)

$$a_i <_{SP} a_j \Leftrightarrow j < i. \quad (3.5)$$

We define the non-strict order to be $a_i \leq_{SP} a_j \Leftrightarrow [a_i = a_j \text{ or } a_i <_{SP} a_j]$. This makes sense because $i \mapsto a_i$ is bijective between \mathcal{R}_d and $\{1, 2, 3, \dots, N\}$. Clearly $<_{SP}$ is a strict total order on \mathcal{R}_d . We call this the *structure preference order* on \mathcal{R}_d , which exists and is unique given $<_D$. This means the transformation $<_D \mapsto <_{SP}$ is functional, where $<_D$ is total on \mathcal{R}_d .

Now let $<_D$ be *any* strict partial order on \mathcal{R}_d . We define the strict set comparison relation on $\mathcal{P}_{\text{fin}}(\mathcal{R}_d)$ corresponding to $<_D$. For $\Gamma, \Gamma' \subseteq_{\text{fin}} \mathcal{R}_d$, the relation \triangleleft_{DEli} , called *the disjoint elitist order*, is defined as follows:

$$\Gamma \triangleleft_{DEli} \Gamma' \Leftrightarrow (\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x <_D y. \quad (3.6)$$

¹⁴We will see this explicitly in Example 1 later.

The lifting $<_D \mapsto \triangleleft_{DEli}$ is functional. We will focus on the following special case of \triangleleft_{DEli} , where instead of $<_D$ we have $<_{SP}$:

$$\Gamma \triangleleft_{SP} \Gamma' \Leftrightarrow (\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x <_{SP} y. \quad (3.7)$$

The corresponding strict argument preference is, for $A, B \in \mathcal{A}$,

$$A \prec_{SP} B \Leftrightarrow DR(A) \triangleleft_{SP} DR(B). \quad (3.8)$$

We define the corresponding non-strict preference as

$$A \lesssim_{SP} B \Leftrightarrow [DR(A) \triangleleft_{SP} DR(B) \text{ or } DR(A) = DR(B)] \quad (3.9)$$

We now show that \lesssim_{SP} satisfies the following properties.

Lemma 3.1. *For all $A, B \in \mathcal{A}$, $DR(A) \subseteq DR(B) \Rightarrow B \lesssim_{SP} A$.*

Proof. If $DR(B) = DR(A)$ then $B \approx A$, so $B \lesssim_{SP} A$. If $DR(A) \subset DR(B)$, then $DR(A) - DR(B) = \emptyset$, which means $B \prec_{SP} A$ is vacuously true from Equation 3.7 so $B \lesssim_{SP} A$. \square

The following result shows that larger arguments, which potentially can contain more fallible information (i.e. defeasible rules), cannot be more preferred than its (smaller) subarguments.

Corollary 3.2. *For all $A, B \in \mathcal{A}$, if $A \subseteq_{arg} B$ then $B \lesssim_{SP} A$.*

Proof. It can be shown from how ASPIC⁺ arguments are constructed (Section 2.2) that $A \subseteq_{arg} B \Rightarrow DR(A) \subseteq DR(B)$, and then invoke Lemma 3.1. \square

Corollary 3.3. *Strict arguments are \lesssim_{SP} -maximal.*

Proof. Let $A \in \mathcal{A}$ be strict and $B \in \mathcal{A}$ be arbitrary. Assume for contradiction that $A \prec_{SP} B$. As $DR(A) = \emptyset$, Equations 3.7 and 3.8 instantiate to: $A \prec_{SP} B \Leftrightarrow (\exists x \in \emptyset) (\forall y \in DR(B)) x <_{SP} y$, which is impossible by the first bounded quantifier. Therefore, if A is strict, then for all $(\forall B \in \mathcal{A}) A \not\prec_{SP} B$. \square

Lemma 3.4. *Let $\langle P, < \rangle$ be a strict toset, then $\langle \mathcal{P}_{fin}(P), \triangleleft_{DEli} \rangle$ is also a strict toset, where \triangleleft_{DEli} is defined in Equation 3.6, here with $<$ instead of $<_D$.*

Proof. We prove \triangleleft_{DEli} is irreflexive, transitive and total on $\mathcal{P}_{fin}(P)$, assuming that $<$ is a strict total order on P .¹⁵ To show irreflexivity, assume for contradiction that there is some $\Gamma \in \mathcal{P}_{fin}(P)$ such that $\Gamma \triangleleft_{DEli} \Gamma$, which by Equation 3.6 is equivalent to a formula whose first bounded quantifier is “ $(\exists x \in \emptyset)$ ”, which

¹⁵ More generally, it can be shown that for *any* strict partial order $<$, the relation \triangleleft_{DEli} from Equation 3.6 is acyclic, and hence irreflexive and asymmetric, but not necessarily transitive. If $<$ is a *modular* order [13, Lemma 3.7], then \triangleleft_{DEli} is transitive. Further, if $<$ total (recalling that total orders are modular), then \triangleleft_{DEli} is trichotomous, and hence a strict total order.

is false, so \triangleleft_{DEli} is irreflexive. To show transitivity, let $n_1, \dots, n_7 \in \mathbb{N}$, such that

$$\begin{aligned} & \{a_1, \dots, a_{n_1}\} \cup \{b_1, \dots, b_{n_2}\} \cup \{c_1, \dots, c_{n_3}\} \cup \{d_1, \dots, d_{n_4}\} \\ & \cup \{e_1, \dots, e_{n_5}\} \cup \{f_1, \dots, f_{n_6}\} \cup \{g_1, \dots, g_{n_7}\} \subseteq P. \end{aligned} \quad (3.10)$$

All of these elements a_1, \dots, g_{n_7} are distinct. If $n_i = 0$ then the corresponding set is empty. Let $\Gamma, \Gamma', \Gamma'' \subseteq_{\text{fin}} P$, where

$$\begin{aligned} \Gamma &= \{a_1, \dots, a_{n_1}\} \cup \{d_1, \dots, d_{n_4}\} \cup \{f_1, \dots, f_{n_6}\} \cup \{g_1, \dots, g_{n_7}\}, \\ \Gamma' &= \{b_1, \dots, b_{n_2}\} \cup \{d_1, \dots, d_{n_4}\} \cup \{e_1, \dots, e_{n_5}\} \cup \{g_1, \dots, g_{n_7}\} \text{ and} \\ \Gamma'' &= \{c_1, \dots, c_{n_3}\} \cup \{e_1, \dots, e_{n_5}\} \cup \{f_1, \dots, f_{n_6}\} \cup \{g_1, \dots, g_{n_7}\}. \end{aligned}$$

We can picture these sets with the Venn diagram in Figure 3.1. The solid outer rectangle represents the set P . The three finite sets $\Gamma, \Gamma', \Gamma''$ are the three rectangles within. Each overlapping region has exactly the elements indicated and nothing more. This configuration exhausts all possibilities for Γ, Γ' and Γ'' .

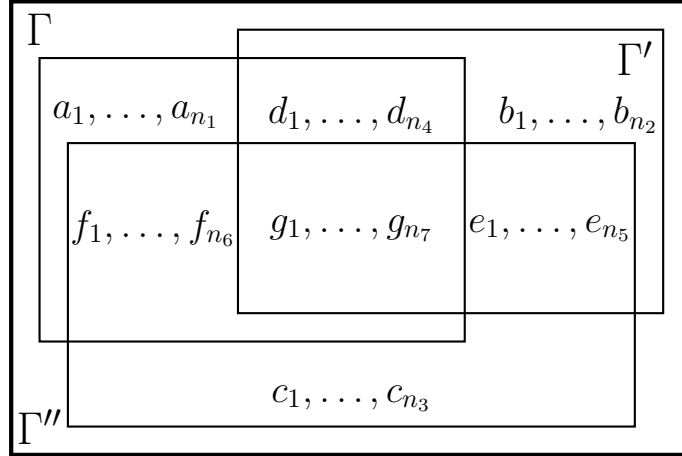


Figure 3.1: Venn diagram for the proof of Lemma 3.4.

Now suppose $<$ permits $\Gamma \triangleleft_{DEli} \Gamma' \triangleleft_{DEli} \Gamma''$, we write this out in terms of elements (Equations 3.11 and 3.12). $\Gamma \triangleleft_{DEli} \Gamma'$ is equivalent to

$$\begin{aligned}
& (\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x < y \\
& \Leftrightarrow (\exists x \in \{a_1, \dots, a_{n_1}\} \cup \{f_1, \dots, f_{n_6}\}) (\forall y \in \{b_1, \dots, b_{n_2}\} \cup \{e_1, \dots, e_{n_5}\}) x < y \\
& \Leftrightarrow (\exists x \in \{a_1, \dots, a_{n_1}\} \cup \{f_1, \dots, f_{n_6}\}) \left[\left(\bigwedge_{i=1}^{n_2} x < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} x < e_j \right) \right] \\
& \Leftrightarrow \bigvee_{k=1}^{n_1} \left[\left(\bigwedge_{i=1}^{n_2} a_k < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \right] \vee \bigvee_{l=1}^{n_6} \left[\left(\bigwedge_{i=1}^{n_2} f_l < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} f_l < e_j \right) \right].
\end{aligned} \tag{3.11}$$

Note that there are $(n_1 + n_6)$ disjuncts in Equation 3.11. Applying the same reasoning as in Equation 3.11, we can see that $\Gamma' \triangleleft_{DEli} \Gamma''$ is equivalent to

$$\bigvee_{k=1}^{n_2} \left[\left(\bigwedge_{i=1}^{n_3} b_k < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_6} b_k < f_j \right) \right] \vee \bigvee_{l=1}^{n_4} \left[\left(\bigwedge_{i=1}^{n_3} d_l < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_6} d_l < f_j \right) \right]. \tag{3.12}$$

There are $(n_2 + n_4)$ disjuncts in Equation 3.11. We need to show that $\Gamma \triangleleft_{DEli} \Gamma''$. By the same reasoning as Equations 3.11 and 3.12, this is equivalent to

$$\bigvee_{k=1}^{n_1} \left[\left(\bigwedge_{i=1}^{n_3} a_k < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \right] \vee \bigvee_{l=1}^{n_4} \left[\left(\bigwedge_{i=1}^{n_3} d_l < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} d_l < e_j \right) \right]. \tag{3.13}$$

So, to prove Equation 3.13, we need to show one of the disjuncts of Equation 3.13 i.e. for at least one of $1 \leq k \leq n_1$ or $1 \leq l \leq n_4$, we show either

$$\left[\left(\bigwedge_{i=1}^{n_3} a_k < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \right] \text{ or } \left[\left(\bigwedge_{i=1}^{n_3} d_l < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} d_l < e_j \right) \right] \tag{3.14}$$

by establishing all of the conjuncts. Given $\Gamma \triangleleft_{DEli} \Gamma' \triangleleft_{DEli} \Gamma''$, we take the conjunction of Equations 3.11 and 3.12, making $(n_1 + n_6)(n_2 + n_4)$ disjuncts, which is equivalent to the following expression:

$$\begin{aligned}
& \left\{ \bigvee_{k=1}^{n_1} \left[\left(\bigwedge_{i=1}^{n_2} a_k < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \right] \vee \bigvee_{l=1}^{n_6} \left[\left(\bigwedge_{i=1}^{n_2} f_l < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} f_l < e_j \right) \right] \right\} \\
& \wedge \left\{ \bigvee_{k=1}^{n_2} \left[\left(\bigwedge_{i=1}^{n_3} b_k < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_6} b_k < f_j \right) \right] \vee \bigvee_{l=1}^{n_4} \left[\left(\bigwedge_{i=1}^{n_3} d_l < c_i \right) \wedge \left(\bigwedge_{j=1}^{n_6} d_l < f_j \right) \right] \right\}.
\end{aligned}$$

As \wedge and \vee bi-distribute, we have four cases:

1. For some $1 \leq k \leq n_1$ and $1 \leq k' \leq n_2$, we have

$$\left(\bigwedge_{i=1}^{n_2} a_k < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \wedge \left(\bigwedge_{i'=1}^{n_3} b_{k'} < c_{i'} \right) \wedge \left(\bigwedge_{j'=1}^{n_6} b_{k'} < f_{j'} \right). \quad (3.15)$$

This means for some $1 \leq k \leq n_1$, we have

$$\left(\bigwedge_{j=1}^{n_5} a_k < e_j \right), \text{ and from} \quad (3.16)$$

$$\left(\bigwedge_{i=1}^{n_2} a_k < b_i \right) \wedge \left(\bigwedge_{i'=1}^{n_3} b_{k'} < c_{i'} \right),$$

that $1 \leq k' \leq n_2$, and transitivity of $<$, we infer

$$\left(\bigwedge_{i=1}^{n_3} a_k < c_i \right). \quad (3.17)$$

Equations 3.16 and 3.17 imply $\Gamma \triangleleft_{DEli} \Gamma''$.

2. For some $1 \leq k \leq n_1$ and $1 \leq l \leq n_4$, we have

$$\left(\bigwedge_{i=1}^{n_2} a_k < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} a_k < e_j \right) \wedge \left(\bigwedge_{i'=1}^{n_3} d_l < c_{i'} \right) \wedge \left(\bigwedge_{j'=1}^{n_6} d_l < f_{j'} \right) \quad (3.18)$$

This case uses the assumption that $<$ is total.¹⁶ The second and the third bracketed conjuncts in Equation 3.18 are necessary but not sufficient to lead to $\Gamma \triangleleft_{DEli} \Gamma''$. Let k_0 be the witness to $1 \leq k \leq n_1$ and l_0 be the witness to $1 \leq l \leq n_4$. As $<$ is total, either $a_{k_0} < d_{l_0}$ or $d_{l_0} < a_{k_0}$ (remember all elements are distinct).

- If $a_{k_0} < d_{l_0}$ then $a_{k_0} < c_i$ for all $1 \leq i \leq n_3$. Therefore, $(\bigwedge_{i=1}^{n_3} a_{k_0} < c_i)$.
- If $d_{l_0} < a_{k_0}$ then $d_{l_0} < e_j$ for all $1 \leq j \leq n_5$. Therefore, $(\bigwedge_{j=1}^{n_5} d_{l_0} < e_j)$.

In either case, $\Gamma \triangleleft_{DEli} \Gamma''$.

3. For some $1 \leq l \leq n_6$ and $1 \leq k \leq n_2$, we have

$$\left(\bigwedge_{i=1}^{n_2} f_l < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} f_l < e_j \right) \wedge \left(\bigwedge_{i'=1}^{n_3} b_k < c_{i'} \right) \wedge \left(\bigwedge_{j'=1}^{n_6} b_k < f_{j'} \right) \quad (3.19)$$

¹⁶It can be shown that if $<$ is not total, \triangleleft_{DEli} is not transitive, see [23, Lemma A.2].

The irreflexivity of $<$ and the first and last bracketed conjuncts gives a contradiction when you run over all indices, so this case gives a contradiction.

4. For some $1 \leq l \leq n_6$ and $1 \leq l' \leq n_4$, we have

$$\left(\bigwedge_{i=1}^{n_2} f_l < b_i \right) \wedge \left(\bigwedge_{j=1}^{n_5} f_l < e_j \right) \wedge \left(\bigwedge_{i'=1}^{n_3} d_{l'} < c_{i'} \right) \wedge \left(\bigwedge_{j'=1}^{n_6} d_{l'} < f_{j'} \right) \quad (3.20)$$

This case is similar to the first case – we use transitivity to combine the second and last bracketed conjuncts. This infers the second conjunct of Equation 3.14, which means $\Gamma \triangleleft_{DEli} \Gamma''$.

Therefore, in all cases, $\Gamma \triangleleft_{DEli} \Gamma''$. This shows \triangleleft_{DEli} is transitive on $\mathcal{P}_{fin}(P)$.

To show trichotomy, let $\Gamma, \Gamma' \in \mathcal{P}_{fin}(P)$ be arbitrary. We start by assuming $\Gamma \neq \Gamma'$ and show exactly one of $\Gamma \triangleleft_{DEli} \Gamma'$ or $\Gamma' \triangleleft_{DEli} \Gamma$ is true. From Equation 3.6, we consider the symmetric difference $\Gamma \ominus \Gamma'$. The set $\langle \Gamma \ominus \Gamma', < \rangle \subseteq \langle P, < \rangle$ is also a finite strict toset. This means there must exist a $<$ -least element $x_0 \in \Gamma \ominus \Gamma'$, say. There are two mutually exclusive possibilities. If $x_0 \in \Gamma - \Gamma'$, then $\Gamma \triangleleft_{DEli} \Gamma'$. If $x_0 \in \Gamma' - \Gamma$, then $\Gamma' \triangleleft_{DEli} \Gamma$. This establishes trichotomy. Therefore, $\langle \mathcal{P}_{fin}(P), \triangleleft_{DEli} \rangle$ is a strict chain. \square

Therefore, given the strict toset $\langle \mathcal{R}_d, <_D \rangle$, $\langle \mathcal{P}_{fin}(\mathcal{R}_d), \triangleleft_{SP} \rangle$ is also a strict toset.

Lemma 3.5. *The argument preference \lesssim_{SP} is a total preorder on \mathcal{A} .*

Proof. We instantiate $\langle P, < \rangle$ in Lemma 3.4 to $\langle \mathcal{R}_d, <_{SP} \rangle$. This is valid because by Equation 3.5 and the discussion afterwards, $<_{SP}$ is a strict total order on \mathcal{R}_d . Further, Equation 3.7 is Equation 3.6 with $<_{SP}$ instead of $<_D$. Therefore, $\langle \mathcal{P}_{fin}(\mathcal{R}_d), \triangleleft_{SP} \rangle$ is a strict toset by Lemma 3.4. By Equation 3.8, \prec_{SP} is a strict total order on \mathcal{A} , and \lesssim_{SP} (Equation 3.9) is a total preorder on \mathcal{A} . \square

By Lemma 3.5, if two arguments A and B satisfy $A \not\prec_{SP} B$, then $B \lesssim_{SP} A$. We demonstrate the features of $<_{SP}$ and \prec_{SP} with Examples 1 and 2.

Example 1. Suppose that instead of respecting the logical structure of the defeasible rules with $<_{SP}$, we use an argument preference relation \prec based on \triangleleft_{DEli} (Equation 3.6) instead of \triangleleft_{SP} , i.e. replace \triangleleft_{SP} in Equation 3.9 with \triangleleft_{DEli} . Now consider the following LPDT. Let T^+ have $W = \emptyset$ and $D = \{d_k\}_{k=1}^5$ where

$$d_1 := \frac{:c_1}{c_1}, d_4 := \frac{c_3 : c_4}{c_4}, d_3 := \frac{:c_3}{c_3}, d_2 := \frac{c_1 : c_2}{c_2}, d_5 := \frac{c_1 : \neg(c_2 \wedge c_4)}{\neg(c_2 \wedge c_4)},$$

such that $d_1 <^+ d_4 <^+ d_3 <^+ d_2 <^+ d_5$. Our PDE is constructed in the usual manner starting from $E_0 = Th(\emptyset)$ by Equation 2.4. Equation 2.5 gives the order of application of the defaults:

$$E_1 = E_0 + c_3, E_2 = E_1 + c_4, E_3 = E_2 + c_1, E_4 = E_3 + \neg(c_2 \wedge c_4), \quad (3.21)$$

with $E_k = E_4$ for all $k \geq 5$. As $\neg(c_2 \wedge c_4) \equiv (\neg c_2 \vee \neg c_4)$, along with c_4 (from d_4), we have $\neg c_2$, which blocks d_2 . The unique PDE from this LPDT is $E := Th(\{c_1, \neg c_2, c_3, c_4\})$. Now consider the corresponding arguments following our instantiation. We have the defeasible rules¹⁷ $r_1 <_D r_4 <_D r_3 <_D r_2 <_D r_5$. The relevant arguments and sets of defeasible rules are

$$\begin{aligned} A &:= [[\Rightarrow c_1] \Rightarrow c_2], DR(A) = \{r_1, r_2\} \\ B &:= [[\Rightarrow c_3] \Rightarrow c_4], DR(B) = \{r_3, r_4\} \\ C &:= [[\Rightarrow c_1] \Rightarrow \neg(c_2 \wedge c_4)], DR(C) = \{r_1, r_5\}, \\ D &:= [B, C \rightarrow \neg c_2], DR(D) = \{r_1, r_3, r_4, r_5\}. \end{aligned}$$

We illustrate these arguments in Figure 3.2. Our convention for diagrams is that broken arrows represent defeasible rules, and solid arrows represent strict rules. Solid vertical lines spanning the length of arguments label those arguments. In the diagrams of this paper, defeasible rules with empty antecedent have the symbol \top as a placeholder for their antecedent.

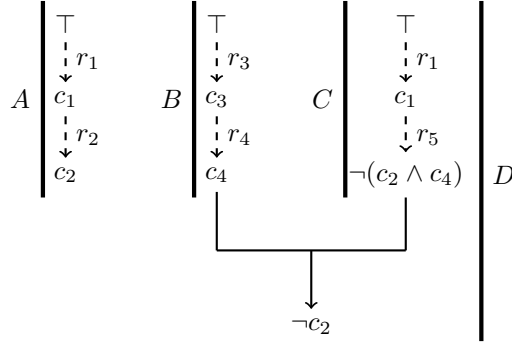


Figure 3.2: The arguments of Example 1.

For the stable extension to correspond to the PDL extension, the desired stable extension contains the arguments $D, B, C, [\Rightarrow c_3], [\Rightarrow c_1]$, the conclusions of which are, respectively, $\neg c_2, c_4, \neg(c_2 \wedge c_4), c_3, c_1$, which under deductive closure, corresponds to E . However, this would require $D \hookrightarrow A$, which means, by Equation 2.3, $D \rhd A$ and $D \not\prec A$. Clearly, $D \rhd A$ on A . However, it is not the case that r_2 is the $<_D$ -least defeasible rule, so $D \prec A$. Therefore, this argument preference relation does not generate the corresponding stable extension to E .

Suppose now that we do respect the logical structure of the rules and use $<_{SP}$ as our argument preference (Equation 3.8). By applying Equations 3.4 and 3.5, we can show that $a_1 = r_3, a_2 = r_4, a_3 = r_1, a_4 = r_5$ and $a_5 = r_2$. The structure preference order is $r_2 <_{SP} r_5 <_{SP} r_1 <_{SP} r_4 <_{SP} r_3$. Notice that

¹⁷We define, for $1 \leq i \leq 5$, $r_i := f(d_i)$, by Equation 3.1.

this is precisely the order in which the corresponding normal defaults are added in PDL, as Equation 3.21 shows. It is easy to show that the corresponding stable extension under the argument preference \prec_{SP} corresponds to the PDL inference, because r_2 is now \prec_{SP} -least, so $D \not\prec_{SP} A$. Therefore $A \prec_{SP} D$, so $A \hookrightarrow D$.

Example 2. However, \prec_{SP} does not necessarily follow the PDL order of the application of defaults. Consider $\langle \{d_1, d_2\}, \{a\}, <^+ \rangle$ with $d_1 := \frac{a:\neg a}{\neg a}$ and $d_2 := \frac{b}{b}$ such that $d_2 <^+ d_1$. We have $E = Th(\{a, b\})$, where d_1 is blocked by W , so d_2 is the only default added. In argumentation, we have $\mathcal{K}_n = \{a\}$, $r_1 := (a \Rightarrow \neg a)$ and $r_2 := (\Rightarrow b)$, such that $r_2 <_D r_1$. The arguments are $A_0 := [a]$, $A_1 := [A_0 \Rightarrow \neg a]$ and $B := [\Rightarrow b]$. Applying Equation 3.4, we have $r_2 \prec_{SP} r_1$, which clearly is not the order of how the corresponding defaults are added in PDL. Yet the correspondence still holds, since $A_0 \hookrightarrow A_1$ because A_0 is strict and strict arguments always defeat any non-strict argument they attack, so the stable extension is the strict extension of $\{A_0, B\}$, the conclusion set of which (after deductive closure) is the extension of the underlying LPDT.

We have now defined the structure-preference argument preference relation \preceq_{SP} . Given an LPDT T^+ , we denote its *attack graph* to be $AG(T^+) := \langle \mathcal{A}, \rightarrow, \preceq_{SP} \rangle$, and its *defeat graph* to be $DG(T^+) := \langle \mathcal{A}, \hookrightarrow \rangle$, where \hookrightarrow is defined by Equation 2.3 with \preceq equal to \preceq_{SP} .

3.3 The Representation Theorem

In this section we state and prove the representation theorem (Theorem 3.14), which guarantees that the inferences with argumentation semantics under the preference \preceq_{SP} correspond exactly to the inferences in PDL.

3.3.1 Non-Blocked Defaults

We introduce some concepts to help prove the representation theorem. Let T be a PDT and $E = \bigcup_{i \in \mathbb{N}} E_i$ one of its extensions generated from the linearisation $<^+ \supseteq <$. The *set of generating defaults (w.r.t. $<^+$)*, $GD(<^+)$, is defined as

$$\begin{aligned} GD_i(<^+) &:= \{d \in D \mid d \text{ is } <^+ \text{-greatest active in } E_i\}, \\ GD(<^+) &:= \bigcup_{i \in \mathbb{N}} GD_i(<^+) \subseteq D. \end{aligned} \tag{3.22}$$

Intuitively, this is the set of defaults applied to calculate E following the priority $<^+$. However, the same E can be generated by distinct total orders.

Example 3. Consider the PDT $\langle \{ \frac{a:c}{c}, \frac{b:c}{c} \}, \{a, b\}, \emptyset \rangle$. We have two possible linearisations $\frac{a:c}{c} <_1^+ \frac{b:c}{c}$ and $\frac{b:c}{c} <_2^+ \frac{a:c}{c}$. By Footnote 8 (page 8) we have $GD(<_1^+) = \{ \frac{b:c}{c} \}$ and $GD(<_2^+) = \{ \frac{a:c}{c} \}$, which are not equal, even though both linearisations give the same extension $E = Th(\{a, b, c\})$. In both cases, the default in $D - GD(<_i^+)$ (for $i = 1, 2$) is not active not because it is blocked by $\neg c$, but rather because it adds no new information.

We wish to distinguish between inactive defaults that conflict with something known and inactive defaults that do not add any new information. We call a default $\frac{\theta:\phi}{\phi}$ *semi-active* (in $S \subseteq \mathcal{SL}$) iff $[\theta \in S, \neg\phi \notin S, \phi \in S]$. Let $<^+ \supseteq <$ be total and which generates the extension E (Equation 2.5). The *set SAD* ($<^+$) of semi-active defaults with respect to the linearisation $<^+$ is defined as

$$\{d \in D \mid d \text{ is semi-active w.r.t. } E, \text{ which is generated by } <^+\}. \quad (3.23)$$

Semi-active defaults add no new information. The *set of non-blocked defaults* is

$$NBD(<^+) := GD(<^+) \cup SAD(<^+) \subseteq D. \quad (3.24)$$

Lemma 3.6. *If $<^+$ generates the PDE E , then*

$$NBD(<^+) := \left\{ \frac{\theta:\phi}{\phi} \in D \mid \theta \in E, \neg\phi \notin E \right\}. \quad (3.25)$$

Proof. It is sufficient to show Equation 3.24 (with Equations 3.22 and 3.23) is the same as the right hand side of Equation 3.25. Let $<^+$ generate the extension E and, for notational convenience, we suppress the argument “ $<^+$ ” in the sets for this proof.¹⁸ (\Rightarrow) Case 1: Assume $d \in SAD$, then $Ante(d) \subseteq E$, $\neg Conc(d) \notin E$ and $Conc(d) \in E$.

$$d \in \{d' \in D \mid Ante(d') \subseteq E, \neg Conc(d') \notin E\} \text{ and hence} \quad (3.26)$$

$$SAD \subseteq \{d \in D \mid Ante(d) \subseteq E, \neg Conc(d) \notin E\}. \quad (3.27)$$

Case 2: Now assume $d \in GD$, which is equivalent to $(\exists i \in \mathbb{N}) d \in GD_i$. This is equivalent to $(\exists i \in \mathbb{N}) [Ante(d) \subseteq E_i, Conc(d) \notin E_i, \neg Conc(d) \notin E_i]$, which is equivalent to

$$Ante(d) \subseteq E_{j_0}, Conc(d) \notin E_{j_0}, \neg Conc(d) \notin E_{j_0} \text{ } j_0 \text{ witness to } i. \quad (3.28)$$

This implies $Ante(d) \subseteq E_{j_0}$, $\neg Conc(d) \notin E_{j_0}$, and it follows that $Ante(d) \subseteq E$, $\neg Conc(d) \notin E_{j_0}$. Clearly, this means $Ante(d) \subseteq E$. Now assume for contradiction that $\neg Conc(d) \in E$, which means there is some $i_0 \in \mathbb{N}$ such that $\neg Conc(d) \in E_{i_0}$. What is the relationship between i_0 and j_0 ? There are three possibilities:

- $i_0 = j_0$ would mean $\neg Conc(d) \notin E_{i_0}$ and $\neg Conc(d) \in E_{i_0}$ – contradiction.
- $i_0 < j_0$: We have $\neg Conc(d) \notin E_{j_0}$ and $\neg Conc(d) \in E_{i_0}$, which is also impossible because the E_i ’s form an ascending chain, so $E_{i_0} \subseteq E_{j_0}$. Therefore, we get $\neg Conc(d) \in E_{j_0}$ and $\neg Conc(d) \notin E_{j_0}$.
- $i_0 > j_0$: We have $\neg Conc(d) \notin E_{j_0}$ and $\neg Conc(d) \in E_{i_0}$. From Equation 3.28, we have that d is active in E_{j_0} , hence $Conc(d) \in E_{j_0+1} \subseteq E_{i_0}$, which makes $\neg Conc(d) \in E_{i_0}$ impossible because the E_i ’s are consistent.

¹⁸For example, instead of writing “ $GD(<^+)$ ” we write “ GD ”.

Therefore, $\neg Conc(d) \notin E$. So we have $Ante(d) \subseteq E$ and $\neg Conc(d) \notin E$. Therefore, Equation 3.26 is true for this case and we have

$$GD \subseteq \{d \in D \mid Ante(d) \subseteq E, \neg Conc(d) \notin E\}. \quad (3.29)$$

We can take the union of Equations 3.27 and 3.29 to get

$$GD \cup SAD \subseteq \{d \in D \mid Ante(d) \subseteq E, \neg Conc(d) \notin E\}. \quad (3.30)$$

(\Leftarrow) Now assume $d \in \{d' \in D \mid Ante(d') \subseteq E, \neg Conc(d') \notin E\}$, which means $Ante(d) \subseteq E$ and $\neg Conc(d) \notin E$. We have, for some $i_0 \in \mathbb{N}$,

$$\begin{aligned} &\Leftrightarrow Ante(d) \subseteq E_{i_0}, (\forall j \in \mathbb{N}) \neg Conc(d) \notin E_j \\ &\Leftrightarrow Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, (\forall j \in \mathbb{N} - \{i_0\}) \neg Conc(d) \notin E_j \\ &\Leftrightarrow (\forall j \in \mathbb{N} - \{i_0\}) \neg Conc(d) \notin E_j \text{ and} \\ &\quad [(Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, Conc(d) \in E_{i_0}) \text{ or} \\ &\quad (Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, Conc(d) \notin E_{i_0})] \\ &\Leftrightarrow (\forall j \in \mathbb{N} - \{i_0\}) \neg Conc(d) \notin E_j \text{ and} \\ &\quad [(Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, Conc(d) \in E_{i_0}) \text{ or } d \in GD_{i_0}] \\ &\Rightarrow (\forall j \in \mathbb{N} - \{i_0\}) \neg Conc(d) \notin E_j \text{ and} \\ &\quad [(Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, Conc(d) \in E_{i_0}) \text{ or } d \in GD] \\ &\Rightarrow d \in GD \text{ or } [Ante(d) \subseteq E_{i_0}, \neg Conc(d) \notin E_{i_0}, Conc(d) \in E_{i_0} \text{ and} \\ &\quad (\forall j \in \mathbb{N} - \{i_0\}) \neg Conc(d) \notin E_j] \\ &\Rightarrow d \in GD \text{ or } [Ante(d) \subseteq E \text{ and } (\forall j \in \mathbb{N}) \neg Conc(d) \notin E_j] \\ &\Rightarrow d \in GD \text{ or } [Ante(d) \subseteq E \text{ and } \neg Conc(d) \notin E] \Leftrightarrow d \in GD \cup SAD. \end{aligned}$$

Therefore, we have

$$\{d \in D \mid Ante(d) \subseteq E, \neg Conc(d) \notin E\} \subseteq GD \cup SAD. \quad (3.31)$$

The result follows from Equations 3.30 and 3.31. \square

Given E , $NBD(<^+)$ is uniquely determined by Equation 3.25, so we will write $NBD(E)$ instead. Equation 3.25 adapts Reiter's idea of a *generating default* [19, page 92, Definition 2] to PDL.

We use these concepts to show that the rearrangement of rules $<_D \mapsto <_{SP}$, as defined in Equation 3.5, does not change the extension of the LPDT. This is because the manner through which $<_{SP}$ incorporates the argument structure captures the idea of Equation 2.5, which is the method of how PDL incorporates both structure and preference when choosing the “ $<^+$ -most active” default.

Lemma 3.7. *Let $T := \langle D, W, <^+ \rangle$ and $T' := \langle D, W, <^{+'} \rangle$ be two LPDTs such that $<^+ \cong <_D \mapsto <_{SP} \cong <^{+'}$,¹⁹ then both T and T' have the same extension E .*

Proof. Let E be the unique extension of T , and E' be the unique extension of T' . To show $E = E'$, we need to show that they have the same generating defaults, i.e. $GD(<^+) = GD(<^{+'})$. As $<^{+'} \cong <_{SP}$, and $<^+ \cong <_D \mapsto <_{SP}$, the rearrangement $<_D \mapsto <_{SP}$ will always choose the $<_D$ -greatest active defeasible rule for a_1 in Equation 3.4, the second $<_D$ -greatest active defeasible rule for $a_2 \dots$ etc. until all defeasible rules are rearranged, but the defeasible rules corresponding to the generating defaults of $<^{+'}$ will always be chosen first in the rearrangement, therefore $GD(<^+) = GD(<^{+'})$ and hence the result follows. \square

3.3.2 Existence and Uniqueness of Stable Extensions

Let T^+ be an LPDT. In this section we show that its defeat graph $DG(T^+)$ has a unique stable extension. We propose an algorithm that imitates how PDL extensions are constructed over an LPDT (Equation 2.5). Given $S \subseteq \mathcal{A}$, $r \in \mathcal{R}_d$, the definition of $Args(\cdot)$ (Equation 2.2) and $<_{SP}$ we define \oplus as $S \oplus r := Args(DR(S) \cup \{r\})$, i.e. we close S under all arguments over all strict rules, all defeasible rules in S , and the addition of a new defeasible rule r .

Consider Algorithm 1, which takes as input the attack graph $AG(T^+)$ of an LPDT T^+ , and the strict chain of defeasible rules under the SP order $\langle \mathcal{R}_d, <_{SP} \rangle$. The output is a set of arguments $S \subseteq \mathcal{A}$. The formal definition is:²⁰

Algorithm 1 Generating a Stable Extension

```

1: function GENERATESTABLEEXTENSION( $\langle \mathcal{A}, \rightarrow, \preceq_{SP} \rangle, \langle \mathcal{R}_d, <_{SP} \rangle$ )
2:    $S \leftarrow \{\text{all strict arguments in } \mathcal{A}\}$ 
3:   for  $r \in \mathcal{R}_d$  from  $<_{SP}$ -greatest to  $<_{SP}$ -smallest do
4:     if  $S \oplus r$  has no attacks,  $(S \oplus r)^2 \cap \rightarrow = \emptyset$ , then
5:        $S \leftarrow S \oplus r$ 
6:   return  $S$ 

```

Algorithm 1 first creates the largest possible set of undefeated arguments that do not attack each other. This includes all strict arguments as they are never defeated nor do they attack each other, and possibly some undefeated defeasible arguments. Then, the algorithm includes the defeasible rules from most to least preferred under $<_{SP}$ and tests whether the resulting arguments that are constructed by the inclusion of such a defeasible rule attack each other in the sense of Equation 3.3 (Lines 4–5). Note that the resulting attack must originate

¹⁹This means that the chain $\langle D, <^+ \rangle$ is order isomorphic to $\langle \mathcal{R}_d, <_D \rangle$ as described by Equations 3.1 and 3.2. Then we calculate $<_{SP}$ from $<_D$ as described in Section 3.2 and form a new chain $\langle D, <^{+'} \rangle$, which is order isomorphic to $\langle \mathcal{R}_d, <_{SP} \rangle$.

²⁰This is a brute-force definition used to prove that stable extensions exist and are unique in such defeat graphs.

from the arguments having at most the defeasible rules added so far. As $<_{SP}$ is total, all defeasible rules are considered, and the result includes as many defeasible rules as possible such that the result has no attacks. Adding the rules in the order of $<_{SP}$ while ensuring no attacks mimics the condition of Equation 2.5. It is clear from the algorithm that S exists and is unique given the input, as it is of the form $Args(R)$ for some $R \subseteq \mathcal{R}_d$ (Equation 2.2). We show S is a stable extension.

Lemma 3.8. *The output S of Algorithm 1 is cf (conflict free).*

Proof. cf is guaranteed by the consistency of \mathcal{K}_n and that defeasible rules $r \in \mathcal{R}_d$ are only added if the resulting arguments do not attack each other (Lines 4 - 5). Therefore, by Equation 2.3, S contains no defeats and must be cf. \square

Note that the setup of the algorithm prevents not just defeats but attacks from appearing in S (Line 4). Normally, this is not sufficient to guarantee that $Conc(S)$ is consistent in FOL.

Example 4. Consider $S = \{[\Rightarrow a], [\Rightarrow b], [\Rightarrow \neg(a \wedge b)]\} \subseteq \mathcal{A}$. There are no attacks in S because attacks are defined syntactically (Equation 3.3, page 10). However, $Conc(S) = \{a, b, \neg(a \wedge b)\}$ is clearly inconsistent in FOL.

We now show that if S has no attacks then $Conc(S)$ is consistent in FOL.

Lemma 3.9. *Let S be the output of Algorithm 1. If $S^2 \cap \neg = \emptyset$ then $Conc(S)$ is consistent.*

Proof. By construction and Lemma 2.1, $Conc(S)$ is deductively closed. Assume for contradiction that $Conc(S)$ is inconsistent, then $\theta, \neg\theta \in Conc(S)$ for some $\theta \in \mathcal{SL}$. Hence there are $A, B \in S$ such that $Conc(A) = \theta$ and $Conc(B) = \neg\theta$. If at least one of $TopRule(A)$ or $TopRule(B)$ are defeasible then at least one of $A \rhd B$ and $B \rhd A$ is the case, hence $S^2 \cap \neg \neq \emptyset$ – contradiction.

Now consider the case where $TopRule(A), TopRule(B) \in \mathcal{R}_s$ are both strict. As $W = \mathcal{K}_n$ is consistent and the rules in \mathcal{R}_s are sound, if A and B are both strict then it cannot be the case they have contradictory conclusions. Therefore, at least one of A and B are defeasible. WLOG say A is defeasible. Suppose by construction $A = [A_1, A_2, \dots, A_i, \dots, A_n \rightarrow \theta]$ and $B = [B_1, B_2, \dots, B_m \rightarrow \neg\theta]$, where $A_i \subseteq_{arg} A$ is defeasible with conclusion a_i (Section 2.2). By closure under transposition of \mathcal{R}_s and the properties of S , we can construct the argument $B^{(i)} := [A_1, A_2, \dots, A_{i-1}, B, A_{i+1}, \dots, A_n \rightarrow \neg a_i]$, and by Lemma 2.1, $B^{(i)} \in S$. If $TopRule(A_i)$ is defeasible, then $B^{(i)} \rhd A_i$ and hence $S^2 \cap \neg \neq \emptyset$ – contradiction, so $TopRule(A_i)$ is not defeasible. As A_i is defeasible we choose some subargument $A'_i \subseteq_{arg} A_i$ and repeat the above line of reasoning for $B^{(i)}$ and A'_i . As all arguments are well-founded, this line of reasoning must terminate at some subargument of A_i whose top rule is defeasible, and hence $S^2 \cap \neg \neq \emptyset$ – contradiction. The result follows. \square

Lemma 3.10. *The set S defeats all arguments outside of itself.*

Proof. Let $R := DR(S)$. Let $B \notin S$ be an arbitrary argument outside of S . We show there is an $A \in S$ such that $A \hookrightarrow B$. Given that $B \notin S$, there must be some rule $r \in DR(B) - R$ that causes S to attack the subargument of B with top rule r , according to Algorithm 1, Line 4. Let $B' \subseteq_{\text{arg}} B$ such that $\text{TopRule}(B') = r$. Let $A \in S$ be the attacker of B' at r , such that $\text{Conc}(A) = \overline{\text{Cons}(r)}$.²¹ This means $A \rightarrow B'$ and hence $A \rightarrow B$. There are two possibilities: either this rule $r \in DR(B)$ is $<_{SP}$ -greatest, or it is not.

If r is $<_{SP}$ -greatest, then $\text{Args}(\emptyset) \oplus r$ contains attacking arguments, so A must be strict and hence $A \hookrightarrow B$. If r is not $<_{SP}$ -greatest, then consider the strict $<_{SP}$ -upper-set of r in $\langle \mathcal{R}_d, <_{SP} \rangle$, $T := \{r' \in \mathcal{R}_d \mid r <_{SP} r'\} \neq \emptyset$. There are two sub-possibilities: either $T \cap R = \emptyset$ or $T \cap R \neq \emptyset$. If $T \cap R = \emptyset$, then adding r to S will create an attack by Algorithm 1, Line 4, and this attack must originate from some $A \in \text{Args}(\emptyset)$ because no rule $<_{SP}$ -larger than r is used in the arguments of S , hence $A \hookrightarrow B$.

If $T \cap R \neq \emptyset$, then adding r to S means its attacker $A \rightarrow B'$ is in $\text{Args}(T \cap R)$.²² Either A is strict or not strict (i.e. defeasible). If it is strict, then $A \hookrightarrow B$ as before. If it is not strict, i.e. $\emptyset \neq DR(A) \subseteq T \cap R$, then by definition $(\forall s \in T) r <_{SP} s$. As $DR(A) \subseteq T \cap R$, we must also have $(\forall s \in DR(A)) r <_{SP} s$. Therefore, there is an $r \in DR(B') - DR(A)$ such that for all rules in $DR(A)$, and hence $DR(A) - DR(B')$, $r <_{SP} s$. By Equation 3.8, we conclude that $B' \prec_{SP} A$, and hence $A \hookrightarrow B'$. Therefore, by definition of \hookrightarrow and \subseteq_{arg} , $A \hookrightarrow B$. \square

Theorem 3.11. *The output of Algorithm 1, S , is a stable extension of $DG(T^+)$.*

Proof. Immediate from Lemmata 3.8 and 3.10. \square

We also have a useful property relating the presence of an argument in a stable extension with its rules, which is independent of Algorithm 1.

Lemma 3.12. *For a LPDT T^+ and defeat graph $DG(T^+)$, if \mathcal{E} is a stable extension of $DG(T^+)$, then $A \in \mathcal{E} \Leftrightarrow DR(A) \subseteq DR(\mathcal{E})$.*

Proof. (\Rightarrow) If $A \in \mathcal{E}$ then $DR(A) \subseteq \bigcup_{A \in \mathcal{E}} DR(A) = DR(\mathcal{E})$ trivially. $(\Leftarrow, \text{contrapositive})$ If $A \notin \mathcal{E}$, then $\mathcal{E} \hookrightarrow A$ at some $A' \subseteq_{\text{arg}} A$. Let $r := \text{TopRule}(A')$. Assume for contradiction that $r \in DR(\mathcal{E})$, then $(\exists B \in \mathcal{E}) r \in DR(B)$, so $\mathcal{E} \hookrightarrow B$ – contradiction, as \mathcal{E} is cf. Therefore, $r \notin DR(\mathcal{E})$. But as $r \in DR(A)$, $DR(A) \not\subseteq DR(\mathcal{E})$. \square

²¹Note that A is appropriately chosen such that $\text{Conc}(A) = \overline{\text{Cons}(r)}$ is syntactic equality. This is always possible because \mathcal{R}_s has all rules of proof of FOL. Therefore, if an argument C concludes θ , and we would want it to conclude ϕ , where $\phi \equiv \theta$, we can just append the strict rule $(\theta \rightarrow \phi) \in \mathcal{R}_s$ to C to create a new argument D that concludes ϕ .

²²We have $A \in \text{Args}(T \cap R)$ because as Algorithm 1 adds the rules one by one according to $<_{SP}$, if adding r to the rules in S and then creating all arguments (with all strict rules) creates an attack, then this attack must be due to some argument whose defeasible rules are amongst $T \cap R$. This is because at the point where the algorithm excludes r , any defeasible arguments constructed then can only have their rules from T .

We have shown that given T^+ and $AG(T^+)$, Algorithm 1 gives a unique output that is a stable extension (Theorem 3.11). We now show that this is the *only* stable extension that $DG(T^+)$ can have.

Theorem 3.13. *Let \mathcal{E} be the stable extension that is the output of Algorithm 1. This is the unique stable extension of $DG(T^+)$.*

Proof. Given $DG(T^+) = \langle \mathcal{A}, \hookrightarrow \rangle$, let \mathcal{E} be the output of Algorithm 1, and assume for contradiction that $\mathcal{E}' \neq \mathcal{E}$ is some other stable extension of $DG(T^+)$. Let $A'_1 \in \mathcal{E}' - \mathcal{E}$. There is an argument $A_2 \in \mathcal{E} - \mathcal{E}'$ such that $A_2 \hookrightarrow A'_1$. There is an argument $A'_3 \in \mathcal{E}' - \mathcal{E}$ such that $A'_3 \hookrightarrow A_2$... and so on. We therefore construct a defeat chain of defeasible arguments

$$\dots \hookrightarrow A'_5 \hookrightarrow A_4 \hookrightarrow A'_3 \hookrightarrow A_2 \hookrightarrow A'_1, \quad (3.32)$$

where all primed arguments belong to \mathcal{E}' and all unprimed arguments are in \mathcal{E} .

In general, suppose $A \hookrightarrow B$, then by Equation 2.3, $A \hookrightarrow C \subseteq_{\text{arg}} B$ for some C , and $A \not\prec_{SP} C$. By Corollary 3.2, $B \lesssim_{SP} C$. Assume for contradiction that $A \prec_{SP} B$, then by Lemma 3.5, $A \prec_{SP} C$ – contradiction, so $A \not\prec_{SP} B$.

Equation 3.32 thus becomes $\dots \not\prec_{SP} A'_5 \not\prec_{SP} A_4 \not\prec_{SP} A'_3 \not\prec_{SP} A_2 \not\prec_{SP} A'_1$. By Lemma 3.5, this is equivalent to $A'_1 \lesssim_{SP} A_2 \lesssim_{SP} A'_3 \lesssim_{SP} A_4 \lesssim_{SP} A'_5 \lesssim_{SP} \dots$. By Equation 3.9 and Lemma 3.12, none of the adjacent arguments in this chain can have the same defeasible rules. This implies $A'_1 \prec_{SP} A_2 \prec_{SP} A'_3 \prec_{SP} A_4 \prec_{SP} A'_5 \prec_{SP} \dots$. The corresponding chain for defeasible rules is, by Equation 3.8, $DR(A'_1) \triangleleft_{SP} DR(A_2) \triangleleft_{SP} DR(A'_3) \triangleleft_{SP} DR(A_4) \triangleleft_{SP} \dots$. As \mathcal{R}_d is a finite set, there are only finitely many possible sets of defeasible rules. This strictly ascending chain must therefore be finite, say of length n . Equation 3.32 must therefore be of finite length, terminating at an undefeated argument B , which may or may not be strict.

$$B \hookrightarrow A'_{n-1} \hookrightarrow \dots \hookrightarrow A_2 \hookrightarrow A'_1 \text{ or } B \hookrightarrow A_{n-1} \hookrightarrow \dots \hookrightarrow A_2 \hookrightarrow A'_1,$$

for some $n \in \mathbb{N}^+$. In the first case, $B \in \mathcal{E} - \mathcal{E}'$ is an undefeated argument, so \mathcal{E}' is not a stable extension – contradiction. In the second case, by similar reasoning, \mathcal{E} is not a stable extension – contradiction. There cannot be another stable extension \mathcal{E}' of $DG(T^+)$, so \mathcal{E} is the unique stable extension of $DG(T^+)$. \square

The defeat graphs $DG(T^+)$ of LPDTs T^+ thus have a unique stable extension.

3.3.3 The Representation Theorem: Statement and Proof

In this section we state and prove the representation theorem which relates the stable extension of $DG(T^+) := \langle \mathcal{A}, \hookrightarrow \rangle$ with the extension of the corresponding LPDT $T^+ := \langle D, W, <^+ \rangle$.

Theorem 3.14. *(The Representation Theorem) Let $AG(T^+)$ be the attack graph corresponding to an LPDT T^+ with defeat graph $DG(T^+)$ under \lesssim_{SP} .*

1. Let E be the extension of T^+ , which is unique (Section 2.3). Then there exists a unique stable extension $\mathcal{E} \subseteq \mathcal{A}$ of $DG(T^+)$ such that $\text{Conc}(\mathcal{E}) = E$.
2. Let $\mathcal{E} \subseteq \mathcal{A}$ be the unique stable extension of $DG(T^+)$ by Theorem 3.13, then $\text{Conc}(\mathcal{E})$ is the extension of T^+ .

Proof. Proof of part 1: To prove the first statement we construct \mathcal{E} in terms of E and show \mathcal{E} is a stable extension of $\langle \mathcal{A}, \hookrightarrow \rangle$. By Theorem 3.13, this stable extension is unique. We finally show $\text{Conc}(\mathcal{E}) = E$.

Given E , we let $\mathcal{E} := \text{Args}(f(NBD(E))) \subseteq \mathcal{A}$, where $NBD(E)$ is defined in Equation 3.25 and $\text{Args}(\cdot)$ is defined by Equation 2.2. This set is unique from the properties of Args . For notational convenience we let $R := f(NBD(E))$. We show \mathcal{E} is a stable extension.

Assume for contradiction that \mathcal{E} is not cf, which means there are arguments $A, B \in \mathcal{E}$ such that $A \hookrightarrow B$, which means $A \rightarrow B$ by Equation 2.3. Let $a := \text{Conc}(A)$. As $A \in \mathcal{E}$, Equation 2.2 and the definition of \mathcal{E} means that $DR(A) \subseteq R$. This means from W and the defaults of $f^{-1}(DR(A)) \subseteq D$ (f is defined by Equation 3.1), which are non-blocked defaults in E , it follows that $a \in E$. As E is deductively closed, this means $\neg a \in E$. Now let $B' \subseteq_{\text{arg}} B$ be the argument such that $\text{TopRule}(B') = (b \Rightarrow \neg a)$ for some appropriate intermediate conclusion $b \in \text{Conc}(\text{Sub}(B))$. As $B \in \mathcal{E}$, this means $(b \Rightarrow \neg a) \in R$. By Equation 3.25, this means $\neg a \notin E$ – contradiction. Therefore, \mathcal{E} is cf.

To show $\text{Args}(R)$ defeats all other arguments, let $B \notin \text{Args}(R)$ be arbitrary. Let $r \in DR(B) - R$ be some rule. Let $B' \subseteq_{\text{arg}} B$ be such that $\text{TopRule}(B') = r$. The rule r corresponds to a default $f^{-1}(r) = \frac{\theta:\phi}{\phi} \notin NBD(E)$ (Equation 3.1). By Equation 3.25, we have two cases: either $\theta \notin E$ or $\neg\phi \in E$.

Case 1: If $\neg\phi \in E$, then we now show there exists an argument $A \in \text{Args}(R)$ such that $A \hookrightarrow B'$ and hence $A \hookrightarrow B$, under \lesssim_{SP} . By Equations 2.4 and 2.5, there is some $i \in \mathbb{N}$ such that $\neg\phi \in E_i$. Suppose $i = 0$ then $W \models \neg\phi$. Compactness means there is some $W' \subseteq_{\text{fin}} W$ such that $W' \models \neg\phi$. We can construct an argument A such that $\text{Prem}(A) = W'$ and $\text{Conc}(A) = \neg\phi$ as there will be appropriate combinations of strict rules in \mathcal{R}_s and premises in \mathcal{K}_n , so $A \rightarrow B$. As $DR(A) = \emptyset \subseteq R$, we must have $A \in \text{Args}(R)$. As A is strict, $A \hookrightarrow B$ is guaranteed by Corollary 3.3 of \lesssim_{SP} .

Now suppose that $i > 0$, then $\neg\phi \in E_j$ where $j > 0$ is the witness for i . Let $d_j \in D$ be the default that is $<^+$ -greatest active in the layer E_j , so the set of defaults that are used in concluding $\neg\phi$ (up to the application of deductive rules) is $S := \{d_0, \dots, d_{j-1}\} \subseteq GD_{j-1}(<^+) \subseteq NBD(E)$. We can construct an argument A such that $\text{Prem}(A) \subseteq W$, $\text{Conc}(A) = \neg\phi$ and $DR(A) = f(S)$. Clearly, $DR(A) = f(S) \subseteq f(NBD(E)) =: R$ and hence $A \in \text{Args}(R)$. It is clear that $A \rightarrow B$, so we need to show $A \not\lesssim_{SP} B$.

Given that $\neg\phi \in E_j$, it must be the case that $\phi \notin E_j$. Therefore, $f^{-1}(r)$ is not $<^+$ -greatest active for all extension layers E_0, \dots, E_{j-1} . Suppose for contradiction that there is some rule $s \in DR(A)$ such that $s <_{SP} r$. Then by Equation 3.4, r must be $<^+$ -greatest active at some E_k for $k < j - 1$, which

would then result in ϕ in E_{k+1} , therefore preventing $\neg\phi \in E_j$ – contradiction.²³ Therefore, r is $<_{SP}$ -smaller than all rules in $DR(A)$. By Equation 3.8, we must have $B \prec_{SP} A$, and hence $A \not\prec_{SP} B$, so $A \hookrightarrow B$. Therefore, for the case of $\neg\phi \in E$, $Args(R)$ defeats all arguments outside it. Therefore, in all cases, if $\neg\phi \in E$, there is some argument $A \in \mathcal{E}$ that defeats B .

Case 2: We will assume $\theta \notin E$, and show that it leads to a contradiction with the method of infinite descent.

If $\theta \notin E$, then there is some proper subargument $B'' \subset_{\text{arg}} B'$ such that $Conc(B'') = \theta$. Since $\theta \notin E$ then it is the case that neither $DR(B'') = \emptyset$ nor in $Args(f(NBD(E)))$. This is because if $DR(B'') = \emptyset$ then $W \supseteq Prem(B'') \models \theta$ so $\theta \in E_0 \subseteq E$ by Equation 2.4. Further, if $B'' \in Args(f(NBD(E)))$ then $f^{-1}(DR(B'')) \subseteq NBD(E)$, so by Equation 3.25, we have $\theta \in E$.

Therefore, as B'' is neither strict nor in $Args(f(NBD(E)))$ there is some other defeasible rule $s \in DR(B'') - f(NBD(E))$. We can repeat the above reasoning with the rule s instead of r : suppose $s = \frac{\theta':\phi'}{\phi}$, then either $\theta' \notin E$ or $\neg\phi' \in E$. In the latter case we can construct an argument $A' \in \mathcal{E}$ concluding $\neg\phi'$ which then defeats B'' as in the case when we assumed $\neg\phi \in E$. In the former case we repeat the reasoning in the previous paragraph, but we cannot do this indefinitely as arguments are well-founded. We will end up with either a proper subargument of B'' or an argument in $Args(R)$ concluding θ . In both cases $\theta \in E$ is true, so assuming $\theta \notin E$ will lead to contradiction by the method of infinite descent.

Therefore, the only reason for $r \notin R$ is because $\neg\phi \in E$. We have shown there is an argument A that defeats any argument containing the rule r . As r belongs to some arbitrary $B \notin \mathcal{E}$, this means $\mathcal{E} := Args(R)$ defeats all arguments outside of it and hence it is a stable extension.

To show that $E = Conc(\mathcal{E})$, we show $E \subseteq Conc(\mathcal{E})$ and $Conc(\mathcal{E}) \subseteq E$. In the first case, let $\theta \in E$ so there is some $i \in \mathbb{N}$ such that $\theta \in E_i$ by Equations 2.4 and 2.5.

If $\theta \in E_0$, we have $W \models \theta$ so by compactness there is some $\Delta \subseteq_{\text{fin}} W$ where $\Delta \models \theta$. Given that \mathcal{R}_s has all rules of proof we can construct a strict argument A such that $Prem_n(A) = \Delta$ and $Conc(A) = \theta$. As strict arguments are undefeated, $A \in \mathcal{E}$ so $\theta \in Conc(\mathcal{E})$.

If $\theta \in E_k$ for some $k \in \mathbb{N}^+$, we can construct a defeasible argument A concluding θ such that $DR(A) \subseteq R$ and hence $A \in \mathcal{E}$, so $\theta \in Conc(\mathcal{E})$. Specifically, we construct an argument whose defeasible rules correspond to the defaults added to E up to E_k .

Conversely, if $\theta \in Conc(\mathcal{E})$ there is an argument in \mathcal{E} concluding θ . If this argument is strict then $\theta \in E_0 \subseteq E$, else, as the defeasible rules are in R then $\theta \in E_k \subseteq E$ for some $k \in \mathbb{N}^+$ that indicates when all of the appropriate defaults needed to conclude θ are included.

Proof of part 2: We show $Conc(\mathcal{E}) \subseteq E$ and $E \subseteq Conc(\mathcal{E})$. For the former, if $\theta \in Conc(\mathcal{E})$ then there is some $A \in \mathcal{E}$ concluding θ . If A is strict

²³For this $s \in DR(A)$, the assumption that $s <_{SP} r$ means that the antecedent of $f^{-1}(r)$ is already in the appropriate extension layer.

then $\theta \in E_0 \subseteq E$. If A is defeasible, then say $DR(A) = \{r_i\}_{i=1}^k$ for some $k \in \mathbb{N}^+$. These defeasible rules do not introduce any inconsistency to \mathcal{E} by Lemma 3.9. Consider the set of corresponding defaults $\{d_i\}_{i=1}^k \subseteq D$ to $DR(A)$. We can choose the smallest index $j \in \mathbb{N}$ such that all of the conclusions of these defaults are included in $E_j \subseteq E$. This is because either $\{d_i\}_{i=1}^k \subseteq GD(E)$, or there is some $d_i \in SAD(E)$, but that would mean $cons(d_i) \in E$ so for some $l \in \mathbb{N}$, $cons(d_i) \in E_l$. Therefore, under deductive closure, $\theta \in E_{j+1} \subseteq E$, so $\theta \in E$. This shows that $Conc(\mathcal{E}) \subseteq E$.

Conversely, let $\theta \in E$, so there is some $i \in \mathbb{N}$ such that $\theta \in E_i$. If $i = 0$, then there is a strict argument A , necessarily in \mathcal{E} as it is undefeated, that concludes θ so $\theta \in Conc(\mathcal{E})$. If $i > 0$, then from $\theta \in E_i$ we can consider the defaults added to E_i , and use the corresponding defeasible rules to construct an argument A such that $Prem(A) \subseteq W$, $Conc(A) = \theta$ and $DR(A)$ to contain exactly those defeasible rules. By definition of $<_{SP}$ and Algorithm 1, these defeasible rules are all present in \mathcal{E} , so $DR(A) \subseteq DR(\mathcal{E})$. By Lemma 3.12, $A \in \mathcal{E}$. Therefore, $\theta \in Conc(\mathcal{E})$. This shows that $E \subseteq Conc(\mathcal{E})$. \square

Theorem 3.14 means that PDL, where the default priority $<$ is a total order, is sound and complete with respect to its argumentation semantics; the inferences of PDL can be formally seen as the conclusions of justified arguments.

Given the definition of $<_{SP}$ and the translation of a LPDT to its defeat graph as described in Section 3.1, we can visualise the representation theorem in the following diagram:

$$\begin{array}{ccc}
 T^+ & \xleftrightarrow[\text{translation}]{} & DG(T^+) \\
 \text{PDE} \downarrow & & \downarrow \text{stable extension} \\
 E & \xleftrightarrow[Conc(\mathcal{E})=E]{} & \mathcal{E}
 \end{array}$$

3.4 Satisfaction of Rationality Postulates

In this section, we prove *directly* that our instantiation of ASPIC⁺ to PDL satisfies the Caminada-Amgoud rationality postulates and hence is normatively rational. We do this by investigating some properties of the stable extension of this defeat graph. Notice that we do not appeal to the sufficient conditions articulated by ASPIC⁺ that, if satisfied, will guarantee normative rationality. We will discuss why in Section 5.

3.4.1 The Stable Extension is Grounded

Lemma 3.15. *Let T^+ be an LPDT with attack graph $AG(T^+)$ and defeat graph $DG(T^+)$. Let $\chi : \mathcal{P}(\mathcal{A}) \rightarrow \mathcal{P}(\mathcal{A})$ be the characteristic function. Let $\mathcal{E} \subseteq \mathcal{A}$ be the stable extension of $DG(T^+)$. Then $(\exists n \in \mathbb{N}^+) \mathcal{E} \subseteq \chi^n(\emptyset)$, where χ^n denotes the n^{th} iterate of χ .*

Before we prove Lemma 3.15, we first establish some notation that will be used in the proof. Recall *interval notation*: for any toset $\langle T, < \rangle$ and $a, b \in T$, define the subsets:

$$\begin{aligned} (a, b) &:= \{x \in T \mid a < x < b\}, [a, b] := \{x \in T \mid a \leq x \leq b\}, \\ [a, b) &:= \{x \in T \mid a \leq x < b\} \text{ and } (a, b] := \{x \in T \mid a < x \leq b\}. \end{aligned}$$

Recall that if $b \leq a$ then $(a, b) = [a, b) = (a, b] = \emptyset$, and if $b < a$ then $[a, b] = \emptyset$. As D is finite let $N := |\mathcal{R}_d|$. Given the LPDT T^+ with defeat graph $DG(T^+)$ and stable extension \mathcal{E} , define the set, for $0 \leq k \leq N$,

$$\mathcal{R}_d - DR(\mathcal{E}) =: \{r_1, \dots, r_k\}, \quad (3.33)$$

where $k = 0$ means $DR(\mathcal{E}) = \mathcal{R}_d$, and $k = N$ means $DR(\mathcal{E}) = \emptyset$. WLOG, we arrange the indices for these rules such that $r_{l+1} <_{SP} r_l$, for $1 \leq l \leq k-1$. This is the set containing the rules that do not feature in \mathcal{E} .

Given the strict toset $\langle \mathcal{R}_d, <_{SP} \rangle$, denote $r_{\max} := \max_{<_{SP}} \mathcal{R}_d$ and similarly for r_{\min} . Both r_{\max} and r_{\min} are uniquely defined as $<_{SP}$ is total. To isolate the defeasible rules that do not make up the arguments in \mathcal{E} , we partition \mathcal{R}_d from smallest to largest in $<_{SP}$ as follows:

$$[r_{\min}, r_k), \{r_k\}, (r_k, r_{k-1}), \{r_{k-1}\}, \dots, \{r_2\}, (r_2, r_1), \{r_1\}, (r_1, r_{\max}]. \quad (3.34)$$

This places the defeasible rules that do not feature in \mathcal{E} into their own singleton sets along the chain $<_{SP}$. These singleton sets contain precisely the defeasible rules skipped over by Algorithm 1 when constructing \mathcal{E} . Note the first and last of these sets may be empty, e.g. when $r_k = r_{\min}$. We name these sets: for $1 \leq i \leq k$, $P_i := \{r_i\}$. Similarly, for $2 \leq i \leq k$, $I_i := (r_i, r_{i-1}) \subseteq \mathcal{R}_d$. We also define $I_1 := (r_1, r_{\max}] \subseteq \mathcal{R}_d$ and $I_{k+1} := [r_{\min}, r_k) \subseteq \mathcal{R}_d$. Equation 3.34 can be written as:

$$I_{k+1}, P_k, I_k, P_{k-1}, \dots, P_2, I_2, P_1, I_1. \quad (3.35)$$

Notice from Equation 3.34 that $\bigcup_{j=1}^{k+1} I_j = \mathcal{R}_d - \{r_k, r_{k-1}, \dots, r_2, r_1\} = DR(\mathcal{E})$.

We define the following counterpart sets of arguments to those in Equation 3.35. For $1 \leq i \leq k$, define

$$AP_i := \{A \in \mathcal{A} \mid DR(A) \cap P_i \neq \emptyset\} \subseteq \mathcal{A}. \quad (3.36)$$

These are the sets of defeasible arguments that have at least one rule excluded from \mathcal{E} . By Lemma 3.12 and the definition of the P_i sets, it is easily shown that for all $1 \leq i \leq k$, $AP_i \cap \mathcal{E} = \emptyset$. Further, for $1 \leq i \leq k+1$,

$$AI_i := \left\{ A \in \mathcal{A} \mid DR(A) \cap I_i \neq \emptyset \text{ and } DR(A) \subseteq \bigcup_{j=1}^i I_j \right\}. \quad (3.37)$$

These are the sets of defeasible arguments where the arguments only have rules from these intervals (as subsets of $DR(\mathcal{E})$), with at least one such rule from the $<_{SP}$ -lowest ranked interval I_i . We also define

$$AI_0 := \{A \in \mathcal{A} \mid DR(A) = \emptyset\} = Args(\emptyset), \quad (3.38)$$

which by Equation 2.2 is the set of all strict arguments. Clearly $AI_0 \subseteq \chi(\emptyset)$.

Lemma 3.16. *It is the case that $AI_1 \subseteq \chi(\emptyset)$.*

Proof. The set AI_1 is cf (conflict free): assume for contradiction that $A, B \in AI_1$ such that $A \hookrightarrow B$. Then as $DR(A), DR(B) \subseteq I_1 \subseteq DR(\mathcal{E})$, by Equations 3.33 and 3.34 and Lemma 3.12, $A, B \in \mathcal{E}$ – contradiction, because \mathcal{E} is cf. Now assume for contradiction that $A \in AI_1$ is defeated, so there is some $B \in \mathcal{A}$ such that $B \hookrightarrow A$. Clearly if $B \in \mathcal{E}$, then \mathcal{E} is not cf because $A \in \mathcal{E}$. Therefore, $B \notin \mathcal{E}$. By Lemma 3.12, $DR(B) \not\subseteq DR(\mathcal{E})$ so for some $1 \leq i \leq k$, $DR(B) \cap P_i \neq \emptyset$. However, by Equation 3.34, there is a rule in $DR(B) - DR(A)$, namely r_i , such that it is $<_{SP}$ -smaller than all rules in $DR(A)$ (and hence $<_{SP}$ -smaller than all rules in $DR(A) - DR(B)$). Therefore, $B \prec_{SP} A$, and hence B cannot defeat A . Therefore, all arguments in AI_1 are also undefeated, so $AI_1 \subseteq \chi(\emptyset)$. \square

For the purposes of the proof of Lemma 3.15, we define, for $i > k + 1$,

$$AI_i = AI_{k+1}. \quad (3.39)$$

Note that the sets of arguments in Equation 3.35 do not partition \mathcal{A} , as an argument can conceivably have rules from two or more of the I_j sets. We now apply these ideas to prove Lemma 3.15.

Proof. (Proof of Lemma 3.15) Given our setup, let \mathcal{E} be the stable extension of $DG(T^+)$. By Theorem 3.13, \mathcal{E} is unique and could only have been constructed by Algorithm 1. Algorithm 1 begins with $Args(\emptyset)$, then adds rules in \mathcal{R}_d from $<_{SP}$ -largest to $<_{SP}$ -smallest as long as the resulting set with the rule contains no arguments attacking each other. From the above notation, it is exactly the rules in the sets P_i that, when included, create arguments that attack each other. This is why these rules in P_i do not feature in \mathcal{E} .

We use strong induction to show that for $i \in \mathbb{N}^+$, $AI_i \subseteq \chi^i(\emptyset)$. The base case, $i = 1$, follows from Lemma 3.16. For the strong inductive step, assume $AI_j \subseteq \chi^j(\emptyset)$ for all $1 \leq j \leq i$. We will show that $AI_{i+1} \subseteq \chi^{i+1}(\emptyset)$. Let $A \in AI_{i+1}$ be arbitrary. This means $DR(A) \subseteq \bigcup_{j=1}^{i+1} I_j$ and $DR(A) \cap I_{i+1} \neq \emptyset$ by Equation 3.37. Either A is defeated by an argument in $\bigcup_{j=1}^i AP_j$ or it is not, where AP_j is defined in Equation 3.36. If A is not defeated by an argument in $\bigcup_{j=1}^i AP_j$, then $A \in \chi(\emptyset)$ as it is undefeated; A cannot be defeated by some argument B in AP_j , for $j > i$, because in that case $B \prec_{SP} A$. As χ is \subseteq -monotonic, $A \in \chi^{i+1}(\emptyset)$. Otherwise, if $\bigcup_{j=1}^i AP_j \hookrightarrow A$, then there is some $1 \leq j \leq i$ such that $AP_j \hookrightarrow A$. Call the witness to j j_0 , so $AP_{j_0} \hookrightarrow A$. Say the

defeating argument is $C \in AP_{j_0}$. But by definition of AP_{j_0} ,

$$\begin{aligned} & \left(\exists B \in \bigcup_{s=0}^{j_0} AI_s \right) B \hookrightarrow C, s=0 \text{ is included as } B \text{ may be strict,} \\ & \Leftrightarrow (\exists 0 \leq s \leq j_0) (\exists B \in AI_s) B \hookrightarrow C \\ & \Rightarrow (\exists 0 \leq s \leq j_0) (\exists B \in \chi^s(\emptyset)) B \hookrightarrow C \text{ by our strong inductive hypothesis.} \end{aligned}$$

This means A is defended by $\chi^s(\emptyset)$, so $A \in \chi^{s+1}(\emptyset)$, for some $0 \leq s \leq j_0$. As $s \leq j_0 \leq i$, this means $s \leq i$ and hence $s+1 \leq i+1$. By \subseteq -monotonicity of χ , $A \in \chi^{i+1}(\emptyset)$. This establishes the inductive step.

However, this proof by induction proves this for all $i \in \mathbb{N}^+$. What happens when $i > k+1$? If $i > k+1$, then by Equation 3.39, $AI_i = AI_{k+1} \subseteq \chi^{k+1}(\emptyset) \subseteq \chi^i(\emptyset)$ and we have no more defeasible rules to add. As the sequence AI_i stabilises the result holds for all $i \in \mathbb{N}^+$ trivially.

Now, as $\emptyset, \chi(\emptyset), \chi^2(\emptyset) \dots$ form an \subseteq -increasing sequence in $\mathcal{P}(\mathcal{A})$, we can take the union of the equations $AI_i \subseteq \chi^i(\emptyset)$ and invoke monotonicity of χ :²⁴

$$\bigcup_{i=1}^{k+1} AI_i \subseteq \bigcup_{i=1}^{k+1} \chi^i(\emptyset) = \chi^{k+1}(\emptyset).$$

We then take the union of both sides with the set of all strict arguments. The left hand side becomes \mathcal{E} . This is because the union of the AI_j sets from $j = 0$ to $k+1$ means any argument in that set cannot have any rules in the P_k sets, and therefore $DR(A) \subseteq DR(\mathcal{E})$ and hence $A \in \mathcal{E}$ by Lemma 3.12.

As the set of all strict arguments is contained in $\chi(\emptyset)$ because they are undefeated, the right hand side stays the same. Therefore, we obtain $\mathcal{E} \subseteq \chi^{k+1}(\emptyset)$, where $0 \leq k \leq N \in \mathbb{N}$ is the number of defeasible rules blocked from \mathcal{E} , which is a natural number.²⁵ This shows the result. \square

Lemma 3.17. *Let $\langle A, \rightarrow \rangle$ be an abstract argumentation framework and χ its characteristic function. Let $G \subseteq A$ be the grounded extension. Then $(\forall n \in \mathbb{N}) \chi^n(\emptyset) \subseteq G$.*

Proof. Immediate by induction on n : χ is \subseteq -monotonic and G is complete. \square

We now instantiate the abstract framework $\langle A, \rightarrow \rangle$ in Lemma 3.17 to the defeat graph $\langle \mathcal{A}, \hookrightarrow \rangle$ of a LPDT.

Corollary 3.18. *Suppose we have an LPDT T^+ with attack graph $AG(T^+) := \langle \mathcal{A}, \rightarrow, \preceq_{SP} \rangle$ and defeat graph $DG(T^+) := \langle \mathcal{A}, \hookrightarrow \rangle$. The characteristic function χ is as usual. The stable extension $\mathcal{E} \subseteq \mathcal{A}$ of $DG(T^+)$ is grounded.*

Proof. From Lemma 3.15, there exists some $n \in \mathbb{N}$ such that $\mathcal{E} \subseteq \chi^n(\emptyset)$. From Lemma 3.17, we have $\chi^n(\emptyset) \subseteq G$, where $G \subseteq \mathcal{A}$ is the grounded extension. But by definition, $G \subseteq \mathcal{E}$ because stable extensions are complete. Therefore, we have, for this $n \in \mathbb{N}$, $G \subseteq \mathcal{E} \subseteq \chi^n(\emptyset) \subseteq G$, so $\mathcal{E} = G$. \square

²⁴Strictly speaking the union should be over all $i \in \mathbb{N}$ but because the AI_i sequence stabilises we only have to care about $0 \leq i \leq k+1$.

²⁵Notice if $k = 0$, there is no conflict, all arguments in \mathcal{E} are undefeated, so $\mathcal{E} \subseteq \chi(\emptyset)$.

3.4.2 The Trivialisation and Rationality Theorems

The trivialisation theorem states that if the underlying default priority is total, then all of Dung's argumentation semantics are equivalent.

Theorem 3.19. *(The Trivialisation Theorem) The defeat graph $\langle \mathcal{A}, \hookrightarrow \rangle$ of an LPDT T^+ has a unique complete extension that is grounded, preferred and stable.*

Proof. Let C be any complete extension of $\langle \mathcal{A}, \hookrightarrow \rangle$, which means C is cf and $\chi(C) = C$. Let G be the grounded extension, then $G \subseteq C$ by definition. As the (unique) stable extension \mathcal{E} is grounded by Corollary 3.18, we have $G = \mathcal{E}$, therefore $\mathcal{E} \subseteq C$. This means either $\mathcal{E} = C$ or $\mathcal{E} \subset C$. In the latter case, there will be some $B \notin \mathcal{E}$ such that $B \in C$, but as $\mathcal{E} \subset C$ is stable, we must have some $A \in \mathcal{E}$ (so $A \in C$) such that $A \hookrightarrow B$. Therefore, C is not cf – contradiction. Therefore, $C = \mathcal{E}$. As \mathcal{E} is unique, C is unique. Therefore, $\langle \mathcal{A}, \hookrightarrow \rangle$ has a unique complete extension that is grounded, stable and hence preferred. \square

We now prove that this instantiation of ASPIC⁺ to PDL satisfies the requirements for normative rationality [10]. Recall that when instantiated to FOL, $Cl_{\mathcal{R}_s}$ becomes deductive closure.

Theorem 3.20. *(The Rationality Theorem) Let $\langle \mathcal{A}, \rightarrow, \preceq_{SP} \rangle$ be the ASPIC⁺ attack graph of PDL and let \mathcal{E} be any of the complete extensions of the corresponding defeat graph $\langle \mathcal{A}, \hookrightarrow \rangle$. Our instantiation satisfies the Caminada-Amgoud rationality postulates.*

Proof. By Theorem 3.13, $\langle \mathcal{A}, \hookrightarrow \rangle$ has a unique stable extension \mathcal{E} , which is a complete and an admissible extension. It is sufficient to prove the postulates for \mathcal{E} because by Theorem 3.19, $\langle \mathcal{A}, \hookrightarrow \rangle$ only has \mathcal{E} as its sole complete extension.

1. To show that \mathcal{E} is subargument closed, recall that Algorithm 1 gives an explicit construction of \mathcal{E} , which is of the form $Args(R)$ for some $R \subseteq \mathcal{R}_d$ which is subargument closed (Equation 2.2).
2. The representation theorem states that $Conc(\mathcal{E}) = E$ and as E is deductively closed, $Conc(\mathcal{E})$ is closed under strict rules.
3. As W is consistent and $Conc(\mathcal{E})$ is the extension, $Conc(\mathcal{E})$ must also be consistent and its deductive closure is consistent.

This shows the result. \square

The rationality theorem establishes that this instantiation of ASPIC⁺ to PDL satisfies all of the Caminada-Amgoud rationality postulates and is a normatively rational instantiation of ASPIC⁺.

Finally, the consistency of \mathcal{E} on the side of PDL allows us to establish a stronger notion of cf for \mathcal{E} on the side of argumentation. This is already implicit in Algorithm 1 Line 4.

Corollary 3.21. *Let $\langle D, W, <^+ \rangle$ be an LPDT with attack graph $\langle \mathcal{A}, \rightarrow, \lesssim_{SP} \rangle$ and corresponding defeat graph $\langle \mathcal{A}, \hookrightarrow \rangle$ that has a unique stable extension \mathcal{E} . We have $\mathcal{E}^2 \cap \rightarrow = \emptyset$, i.e. no two arguments in \mathcal{E} attack each other.*

Proof. Given the hypotheses, assume for contradiction that $A \rightarrow B$ for $A, B \in \mathcal{E}$. WLOG we can assume that $\text{Conc}(A) = \theta$ and $\text{Conc}(B) = \neg\theta$ with $\text{TopRule}(B) \in \mathcal{R}_d$, by Equation 3.3 and that \mathcal{E} is subargument closed. Therefore, $\text{Conc}(\mathcal{E})$ is inconsistent, because $\theta, \neg\theta \in \text{Conc}(\mathcal{E})$. This violates the Rationality Theorem – contradiction. Therefore, no two arguments in \mathcal{E} attack each other. \square

3.4.3 Inconsistent Arguments

We have stated in Section 2.2 that arguments are constructed freely from the premises and rules. In this instantiation, it is possible to construct arguments that are inconsistent, either in their intermediate conclusions or their conclusion.

Example 5. Consider the rules $(\Rightarrow a)$ and $(\Rightarrow \neg a)$ and arguments $A = [\Rightarrow a]$, $B = [\Rightarrow \neg a]$ and $C = [A, B \rightarrow \perp]$. Then C is inconsistent in its conclusion. Further, for any $c \in \mathcal{SL}$ the argument $A^+ := [A \rightarrow (a \vee c)]$, so given that our strict rules are the rules of proof in FOL, we can construct the argument $D = [A^+, B \rightarrow c]$ for any c . The intermediate conclusions of D are inconsistent.

We call an argument *inconsistent* iff $\text{Conc}(\text{Sub}(A)) \subseteq \mathcal{FL}$ is an inconsistent set in FOL. It is possible to construct such arguments in \mathcal{A} . We can ignore these arguments by focussing on $\mathcal{A} - \{A \in \mathcal{A} \mid A \text{ is inconsistent}\}$ and restricting \rightarrow in the usual way, but this seems inelegant especially when the process of argumentation is meant to resolve inconsistencies. If we do include inconsistent arguments, the very least is that they should not be justified. By Theorems 3.13 and 3.19, there is only one way of justifying arguments: an argument A is *justified* iff $A \in \mathcal{E}$. Lemma 3.9 and Corollary 3.21 ensure that if A is inconsistent, then $A \notin \mathcal{E}$.

In summary, although it is possible to have inconsistent arguments in \mathcal{A} , they can never be justified and we do not need to be concerned with them.

3.5 Summary

In this section, we have provided an argumentative characterisation of PDL inference that is sound, complete and normatively rational, in the case where our default priority is a strict total order. We can construct ASPIC⁺ arguments and attacks (Section 3.1). The structure-preference relation, \lesssim_{SP} , takes into account both the default priority $<$ and the logical structure of arguments. This is motivated by how PDL adds defaults when constructing extensions (Section 3.2). The representation theorem states that under \lesssim_{SP} , the PDL extension and the conclusion set of the stable extension correspond exactly (Section 3.3, Theorem 3.14). We can prove directly that the stable extension of interest satisfies the Caminada-Amgoud rationality postulates (Section 3.4, Theorem 3.20). As this is the only complete extension of our defeat graphs, our instantiation satisfies

the postulates. Finally, we do not need to explicitly prevent the construction of inconsistent arguments, because they are never justified (Section 3.4.3).

4 On Lifting the Assumption of a Total Order Default Priority

In Section 3, we have provided an argumentative characterisation of PDL inference where the default priority $<$ is a strict total order. It seems that we have lost generality but this is not the case because calculating an extension in PDL always presupposes a linearisation $<^+$ of $<$ (Section 2.3), and Theorem 3.14 shows that for *any* such linearisation the correspondence of inferences between PDL and its argumentation semantics holds.

But argumentation can also define argument preference relations based on an underlying *partial* order. We now investigate how to lift the assumption that $<$ is total for the LPDT T^+ , such that the resulting multiple stable extensions each correspond to an extension of the underlying PDT T . Our underlying representation of PDL in ASPIC⁺ is the same as in Section 3.1, but now $<_D$ is a strict *partial* order.

4.1 The Argument Preference Relation based on Partial Order Default Priorities

In Section 3.2, we devised the structure-preference (SP) argument preference relation \lesssim_{SP} which captures the PDL idea of adding the “ $<^+$ -greatest active” default (Equation 2.5). If we translate a PDT T directly into an argument graph $AG(T)$ without first linearising $<$, the generalised version of \lesssim_{SP} should take into account the incomparabilities of rules while still respecting their logical structure. We formalise this idea by defining a string representation of the rules that will be used in an algorithm to calculate $<_D \mapsto <_{SP}$ for partial $<_D$. We will give a variation of the Penguin Triangle (Example 8) as a running example.

4.1.1 A Representation of Rules and their Ordering using Strings

Let *rulenames* be a set of characters, with as many characters as there are rules in \mathcal{R}_d . Let $g : \mathcal{R}_d \rightarrow \textit{rulenames}$ be a bijection such that each $r \in \mathcal{R}_d$ has a single-character name²⁶ $g(r)$. Let \star denote the Kleene star and $*$ denote string concatenation,²⁷ and $len : \textit{rulenames}^* \rightarrow \mathbb{N}$ returns the number of letters of the string. We will also assume that in each string $\sigma \in \textit{rulenames}^*$ there is an index associated with each letter starting from 0 and ending in $len(\sigma) - 1$. To iterate over the letters l of the string σ we will write $l \in \sigma$, which starts from the

²⁶By “name” we do not mean the naming function $n : \mathcal{R}_d \rightarrow \mathcal{L}$ in Section 2.2 (page 4), which is still undefined ($n \equiv *$) in this case, but just what we label the rules with, e.g. the defeasible rule $r_7 = (a \Rightarrow b)$ has the label or name r_7 .

²⁷This is abuse of notation as we had earlier stated $*$ refers to undefined quantities (Section 2.1). But there are few undefined quantities and the meaning of $*$ will be clear from context.

letter at index 0 and terminates at the letter at index $\text{len}(\sigma) - 1$. The *empty string* is $\varepsilon := ""$ with $\text{len}(\varepsilon) = 0$. We may put quotation marks around strings to emphasise that they are strings.

For $R \subseteq \mathcal{R}_d$ such that $R := \{r_1, \dots, r_k\}$, we can form the string $g(r_1) * g(r_2) * \dots * g(r_k)$, written $g(r_1)g(r_2) \dots g(r_k)$. Notice that forming a string from a set imposes an order on the elements.

Example 6. (Example 1 continued) Suppose we have $r_i := f(d_i)$ for f as in Equation 3.1. Suppose $g(r_1) = "a"$, $g(r_2) = "b"$, $g(r_3) = "c"$, $g(r_4) = "d"$ and $g(r_5) = "e"$. Then for the set $S = \{r_1, r_2, r_5\}$ we can form the strings (e.g.) *"abe"* or *"bea"*, depending on which order we choose the rules to be in.

As g is a bijection we can define the reverse process. Suppose we have a string σ . We define the set of rules that are encoded by the letters of σ as follows:

$$\bigcup_{l \in \sigma} \{g^{-1}(l)\}. \quad (4.1)$$

Notice that we lose the information about the index, but we will see that it does not matter. Notice also that if $\sigma = \varepsilon$ then we have the empty union so the set of rules encoded by ε is \emptyset .

Example 7. (Example 6 continued) Suppose we want to find the set of the string *"ace"*. Applying Equation 4.1, we get the set $\{r_1, r_3, r_5\}$.

Lastly, we can transform strings into total orders: for $\sigma = \sigma_1 \sigma_2 \dots \sigma_n$, where for $1 \leq i \leq n$ the letter σ_i has index $i - 1$.²⁸ We can transform σ to the set

$$\{(\sigma_1, \sigma_2), (\sigma_1, \sigma_3), \dots, (\sigma_1, \sigma_n), (\sigma_2, \sigma_3), \dots, (\sigma_{n-1}, \sigma_n)\} \quad (4.2)$$

using two nested loops ranging over the letters of σ such that the pair of letters (σ_i, σ_j) is added to the set iff $i < j$. Intuitively, given a selected letter σ_i of a string, letters to the right of σ_i are larger than σ_i , and letters to the left of σ_i are smaller than σ_i .

4.1.2 Algorithm and Example Calculation

We want to generalise the mapping $<_D \mapsto <_{SP}$, defined in Section 3.2 for the case where $<_D$ is total, to arbitrary partial orders $<_D$. Furthermore, we want to simultaneously capture all possible linearisations of $<_D$. We now present the algorithm that calculates the generalised mapping $<_D \mapsto <_{SP}$ in two parts.

The first stage of the algorithm is a non-recursive depth first search algorithm that returns the set of all strings representing the rules chosen in accordance with both the preference and the structure as described in Section 4.1.1. This is articulated in Algorithm 2, which defines the function `STRUCTUREPREFERENCE1`. This function takes $\langle \mathcal{R}_d, <_D \rangle$ as input and returns this set of strings.

²⁸Here, σ_i denotes one letter; the subscript i is not a separate letter to σ itself.

The second stage of the algorithm is to turn the output of Algorithm 2 into $<_{SP}$. This is done by translating each string in the output of Algorithm 2 into a strict total order on \mathcal{R}_d , and then taking their intersection. This is articulated in Algorithm 3, which defines a function `STRUCTUREPREFERENCE2`, which takes as input a set of strings, and calculates $<_{SP}$.

Algorithm 2 Calculating $<_{SP}$ from $<_D$ on \mathcal{R}_d , Part 1 – generate a set of strings, each string is an order of the choice of rules from least preferred (the first letter, on the left) to the most preferred (the last letter, on the right), which essentially corrects $<_D$ for the argument structure and then linearises. Recall that in Line 11, $*$ refers to string concatenation. As \mathcal{R}_d is finite, Algorithm 2 terminates. Throughout, S_i is a set of strings, while $R^\lambda, T^\lambda \subseteq \mathcal{R}_d$.

```

1: function STRUCTUREPREFERENCE1( $\langle \mathcal{R}_d, <_D \rangle$ )
2:    $S_0 \leftarrow \{\varepsilon\}$ 
3:    $N \leftarrow |\mathcal{R}_d|$ 
4:   for  $i = 0$  to  $N$  do
5:      $S_{i+1} \leftarrow \emptyset$ 
6:     for  $\lambda \in S_i$  do
7:        $T^\lambda \leftarrow \bigcup_{l \in \lambda} \{g^{-1}(l)\}$ 
8:        $R^\lambda \leftarrow \max_{<_D} [\{s \in \mathcal{R}_d \mid \text{Ante}(s) \subseteq \text{Conc}(\text{Args}(T^\lambda))\} - T^\lambda]$ 
9:       if  $R^\lambda = \emptyset$  then return  $S_i$ 
10:      for  $t \in g(R^\lambda)$  do
11:         $S_{i+1} \leftarrow S_{i+1} \cup \{t * \lambda\}$ 

```

The intuition of Algorithm 2 is as follows: we initialise the algorithm (Line 2) and iterate $N + 1$ times (Line 4), where $N = |\mathcal{R}_d|$ (Line 3). At each iteration Algorithm 2 chooses all of the most preferred applicable rules at that stage. Each choice may render more rules active for the next iterations. It iterates over all such possibilities and repeats this process until the N^{th} iteration, where there are no more rules to be chosen and the algorithm terminates. The result is a set of strings, which are read from right to left, where the right-most letter is the first choice of most preferred applicable defeasible rule, and the left-most letter is the last choice, which usually corresponds to a blocked default.

The intuition of Algorithm 3 is that upon input of this set of strings, the algorithm turns each string into a strict total order over \mathcal{R}_d as described in the end of Section 4.1.1 (Equation 4.2), and then takes the intersection of all such orders to return $<_{SP}$. The intersection returns the “core” strict partial order which is the “smallest” change to the original $<_D$ that is compatible with all argument structures. Given $\langle \mathcal{R}_d, <_D \rangle$, we define:

$$\begin{aligned}
<_{SP} &:= \text{STRUCTUREPREFERENCE2}(\text{STRUCTUREPREFERENCE1}[\langle \mathcal{R}_d, <_D \rangle]) \\
&:= F(<_D).
\end{aligned} \tag{4.3}$$

This is our method for calculating $\langle \mathcal{R}_d, <_{SP} \rangle$ from $\langle \mathcal{R}_d, <_D \rangle$ where $<_D$ is partial.

Algorithm 3 Calculating $<_{SP}$ from $<_D$ on \mathcal{R}_d , Part 2 – from the set of strings generated from Algorithm 2, we turn each string into a strict total order on \mathcal{R}_d , and then take their intersection to return $<_{SP}$.

```

1: function STRUCTUREPREFERENCE2( $S$ )
2:    $orders \leftarrow \emptyset$ 
3:   for  $\lambda \in S$  do
4:      $<_{SP}^+ \leftarrow \emptyset$ 
5:     for  $r \in \lambda$  do
6:       for  $s \in \lambda$  do
7:         if  $index(r) < index(s)$  then
8:            $<_{SP}^+ \leftarrow <_{SP}^+ \cup \{(r, s)\}$ 
9:    $orders \leftarrow orders \cup \{<_{SP}^+\}$ 
return  $\bigcap orders$ 

```

Example 8. (Modified Penguin Triangle) Let $W = \emptyset$,

$$D = \left\{ d_1 := \frac{:a}{a}, d_2 := \frac{a:b}{b}, d_3 := \frac{: \neg b}{\neg b} \right\}$$

and consider the default priority $\leq = \{(d_3, d_2)\}$.²⁹ There are three possible linearisations of $<$ giving two possible extensions:

$E_1 := Th(\{a, b\})$ from $d_3 <^+ d_2 <^+ d_1$ and $d_3 <^+ d_1 <^+ d_2$, and

$E_2 := Th(\{a, \neg b\})$ from $d_1 <^+ d_3 <^+ d_2$.

Let $r_i := f(d_i)$ for $i = 1, 2, 3$ (Equation 3.1). We illustrate these arguments in Figure 4.1.

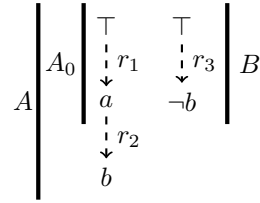


Figure 4.1: The arguments of Example 8.

Clearly A and B attack each other at their conclusions. Putting $<_D = \{(r_3, r_2)\}$ into Equation 4.3, we get the following:

- For Algorithm 2, we have $S_0 = \{\varepsilon\}$, $N = 3$ and $i = 0, 1, 2, 3$.

When $i = 0$, $S_1 = \emptyset$, $\lambda = \varepsilon$, $T^\lambda = \emptyset$ and $R^\lambda = \{r_1, r_3\} \neq \emptyset$. Therefore, $S_1 = \{“r_1”, “r_3”\}$. Notice r_3 is a $<_D$ -maximal applicable rule, because even though $r_3 <_D r_2$, r_2 is not applicable until r_1 is applied.

²⁹Notice this is not the “usual” partial order priority that respects the specificity principle.

When $i = 1$, $S_2 = \emptyset$, and either $\lambda = "r_1"$ or $\lambda = "r_3"$. The former gives $S_2 = \{ "r_2r_1" \}$ and the latter gives $S_2 = \{ "r_2r_1", "r_1r_3" \}$.

When $i = 2$, $S_3 = \emptyset$, $\lambda = "r_2r_1"$ or $\lambda = "r_1r_3"$. In the former case, $S_3 = \{ "r_3r_2r_1" \}$, and in the latter case, $S_3 = \{ "r_3r_2r_1", "r_2r_1r_3" \}$.

When $i = 3$, we get $T^\lambda = \mathcal{R}_d$ so $R^\lambda = \emptyset$, halting the algorithm with output S_3 .

- Given S_3 as input to Algorithm 3, we get the intersection of the words $"r_3r_2r_1"$ and $"r_2r_1r_3"$ when converted to chains, giving $<_{SP} = \{(r_2, r_1)\}$.

Therefore, Equation 4.3, returns $<_{SP} = F(<_D) = \{(r_2, r_1)\}$. If $<_D$ is arbitrary,³⁰ we can repeat the above calculation and obtain the values of Equation 4.3,³¹ which are shown in Table 1.

Values of input $<_D$	Output $<_{SP} = F(<_D)$
(12,32), 32, 12, \emptyset , 21	21
123, 132, (13,23), (12,13), 13, 213	213
31, (21,31)	(21,31)
23, (21,23)	(21,23)
312, (31,32), 321	321
231	231

Table 1: The values of $<_{SP}$ given all possible $<_D$ of Example 8.

4.1.3 Properties of the Generalised SP Order

We prove some properties of F (Equation 4.3 and Algorithms 2 and 3) that will be useful in proving the representation theorem for the case where $<_D$ is a partial order (Theorem 4.11). It can easily be shown that F is a well-defined function from $PO(\mathcal{R}_d)$ to itself, where $PO(\mathcal{R}_d)$ is the set of all strict partial orders on \mathcal{R}_d (Section 2.1). We show that the function $F : PO(\mathcal{R}_d) \rightarrow PO(\mathcal{R}_d)$ indeed generalises the definitions in Section 3.2. Recall that $TO(X)$ is the set of all strict total orders on the set X .

Lemma 4.1. *If $<_D \in TO(\mathcal{R}_d)$ then we recover $<_{SP}$ defined in Section 3.2.*

Proof. If $<_D$ is total, then R^λ on Algorithm 2 Line 8 is singleton. This means Line 10 has only one choice in R^λ , so S_i for all $0 \leq i \leq N$ is singleton. Using the notation of Equation 3.4, Algorithm 2 returns $\{ "a_N a_{N-1} \dots a_2 a_1" \}$. This gets transformed into Equation 3.5 through Algorithm 3, which is $<_{SP}$ for the case of $<_D$ total. \square

³⁰There are 19 partial orders on a set of three labelled elements.

³¹ We abbreviate the total order $r_3 <_D r_2 <_D r_1$ as 321, $<_D := \{(r_1, r_2), (r_3, r_2)\}$ as (12,32), and $<_D = \{(r_1, r_3)\}$ as 13... etc. and the same applies to $<_{SP}$.

Example 9. (Example 8 continued) By restricting F to $TO(\mathcal{R}_d)$, we obtain the following subtable of Table 1, which indeed generalises the calculation for the case where the input $<_D$ is total.

Values of input $<_D$	Output $<_{SP} = F(<_D)$
123, 132, 213	213
312, 321	321
231	231

Table 2: The values of $<_{SP}$ given the linear $<_D$ of Example 9.

The next result shows that given the input $<_D$ in Algorithm 2, each string in the output set of Algorithm 2, when transformed into its corresponding total order on \mathcal{R}_d , is the output of Equation 4.3 for some linearisation $<_D^+$ of $<_D$.

Theorem 4.2. *Consider Algorithm 2 with input $<_D \in PO(\mathcal{R}_d)$. For each string σ in the output set of Algorithm 2, let $<_{out}$ denote σ transformed into a strict total order on \mathcal{R}_d (Algorithm, 3). For each such $<_{out}$ there exists a linearisation $<_D^+$ of $<_D$ such that $<_{out} = F(<_D^+)$.*

Proof. If $<_D$ is itself total, then the output of Algorithm 2 is singleton, which when converted to a chain by Algorithm 3 gives $<_{out} = <_{SP}$. Therefore, there exists a linearisation of $<_D$, namely itself, such that $<_{out} = F(<_D)$.

If $<_D$ is not total, then incomparable rules will cause R^λ (Algorithm 2 Line 8) to not be singleton. Each element of R^λ will form a distinct element of the output set of Algorithm 2. Choosing a given rule r in R^λ to append to the string can also be interpreted as a resolution of this incomparability of $<_D$ through a linearisation $<_D^+$ of $<_D$ that ranks r higher than the alternative choices. Reasoning in this way in all cases whenever R^λ is not singleton, we obtain a linearisation $<_D^+$ of $<_D$ such that $F(<_D^+)$ corresponds to one of the elements in the output S_N of Algorithm 2. \square

It follows that Equation 4.3 incorporates all possible linearisations of $<_D$ in the following manner.

Corollary 4.3. *The output of Equation 4.3 is equal to*

$$<_{SP} = F(<_D) = \bigcap_{<_D^+ \supseteq <_D \text{ total}} F(<_D^+),$$

where the intersection ranges over all linearisations of $<_D$.

Proof. This is immediate from the definition of Algorithm 3 and Theorem 4.2. \square

Example 10. (Example 9 continued) Consider $<_D = 23$, which abbreviates $<_D = \{(r_2, r_3)\}$. 23 has linearisations abbreviated as 123, 213 and 231 (see Footnote 31, page 37). By Table 2, these input linearisations returns, respectively, 213, 213 and 231. By Corollary 4.3, we get $<_{SP}$ to be the intersection of the sets representing the total orders 213 and 231. This gives (21, 23), which abbreviates $<_{SP} = \{(r_2, r_1), (r_2, r_3)\}$. This is consistent with Table 2.

We now relate the linearisations of the inputs and outputs of Table 1.

Theorem 4.4. (*The linearisation square*) Let $<_D \in PO(\mathcal{R}_d)$ and $<_D^+$ be a linearisation of $<_D$. $F(<_D^+) =: <_{SP}^+$ is a linearisation of $F(<_D) =: <_{SP}$.

Proof. Let $<_{SP}$ be as given. Let $<_D^+$ be a linearisation of $<_D$. Suppose $<_D^+$ is the input of Algorithm 2. This will give an output set consisting of a single string that Algorithm 3 translates into some strict linear order $<_{SP}^+$ (say), by Theorem 4.1. Upon input $<_D$ to Algorithm 2, the string corresponding to $<_{SP}^+$ will appear in the output set of Algorithm 2, because we can choose the rules in R^λ (Line 8) in accordance with the ranking of $<_D^+ \supseteq <_D$. By Algorithm 3, $<_{SP}^+ \supseteq F(<_D) = <_{SP}$. \square

The linearisation square can be expressed in the following commutative diagram:

$$\begin{array}{ccc}
 <_D & \xrightarrow[\text{linearisation}]{} & <_D^+ \\
 \text{Equation 4.3} \downarrow & & \downarrow \text{Equation 4.3} \\
 F(<_D) & \xrightarrow[\text{linearisation}]{} & F(<_D^+)
 \end{array}$$

The linearisation square states that the function $F : PO(\mathcal{R}_d) \rightarrow PO(\mathcal{R}_d)$ preserves linearisations.

Example 11. (Example 10 continued) Consider Table 1 again. Let $<_D = 31$. Let $<_D^+ = 321$. We know that $<_{SP} = (21, 31)$. We also know that $<_{SP}^+ = 321$. Clearly, 321 is a linearisation of (21, 31).

4.1.4 The Generalised Argument Preference Relation for $<_D$ Partial

The following result states that changing the partial order PDL default priority $<$ to respect the logical dependencies of defaults while following the preference does not change the PDL extension. This generalises Lemma 3.7 (page 21) to the case where $<_D$ is not necessarily total.

Lemma 4.5. Let $T := \langle D, W, < \rangle$ and $T' := \langle D, W, <' \rangle$ be two PDTs such that $< \cong <_D$ and $F(<_D) \cong <'$, then both PDTs have the same extensions

Proof. Denote $Ext(T)$ and $Ext(T')$ to be the sets of extensions of the respective PDTs. We show that $Ext(T) = Ext(T')$.

(\Rightarrow) Let $E \in Ext(T)$ be arbitrary. This means E is the unique extension of some LPDT $T^+ := \langle D, W, <^+ \rangle$, where $<^+ \supseteq <$ is a strict total order. Therefore, $<^+ \cong <_D^+ \supseteq <_D \cong <$, where $<_D^+$ is a linearisation of $<_D$. By the linearisation square (Theorem 4.4), $F(<_D^+) =: <_{SP}^+$ is a linearisation of $F(<_D) =: <_{SP}$. As $<' \cong <_{SP}$, then $<_{SP}^+ \cong <'^+$, where $<'^+$ is some linearisation of $<'$. By Lemma 3.7, E is also the unique extension of the LPDT $\langle D, W, <'^+ \rangle$, which means E is an extension of $T' = \langle D, W, <' \rangle$. Therefore, $E \in Ext(T')$.

(\Leftarrow) Let $E \in Ext(T')$ be arbitrary. This means E is the unique extension of some LPDT $T^+ := \langle D, W, <'^+ \rangle$, where $<'^+ \supseteq <'$ is a strict total order. As $<'^+ \cong <_{SP}^+$, which is a linearisation of $<_{SP}$, then by Theorem 4.2, there exists a linearisation $<_D^+$ of $<_D$ such that $<_D^+ \mapsto <_{SP}^+$, given that $<_D \mapsto <_{SP}$. By Lemma 3.7, E is the unique extension of $T^+ := \langle D, W, <^+ \rangle$, where $<^+ \cong <_D^+$, which means E is an extension of T . Therefore, $E \in Ext(T)$. Therefore, $Ext(T) = Ext(T')$. \square

Example 12. (Example 11 continued) Recall the setup of Example 8, where $W = \emptyset$ and

$$D = \left\{ d_1 := \frac{:a}{a}, d_2 := \frac{a:b}{b}, d_3 := \frac{: \neg b}{\neg b} \right\}.$$

Consider two strict partial orders on D , $<$ and $<'$, where $d_1 < d_2$ only and $d_2 <' d_1$ only. This gives us two PDTs $T = \langle D, W, < \rangle$ and $T' = \langle D, W, <' \rangle$. Let $<_D \cong <$. By Table 1, $F(<_D) = 21$ so $F(<_D) \cong <'$. Both PDTs T and T' have the same extensions. In the case of T , we have linearisations 312, 132 and 123, with the first linearisation giving $E_1 := Th(\{a, b\})$ and the latter two linearisations giving $E_2 := Th(\{a, \neg b\})$. In the case of T' , we have linearisations 321, 231 and 213, with the first linearisation giving E_1 and the latter two linearisations giving E_2 . Therefore, both T and T' have the same extensions.

We can now define the associated set comparison relation from this new $<_{SP}$ just like Equation 3.7: for $\Gamma, \Gamma' \subseteq_{\text{fin}} \mathcal{R}_d$,

$$\Gamma \triangleleft_{SP} \Gamma' \Leftrightarrow (\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x <_{SP} y, \quad (4.4)$$

where given the partial order default priority $<$, order isomorphic to $<_D$ (Equation 3.1), $<_{SP} = F(<_D)$ is the output of Equation 4.3. The associated strict argument preference is, for $A, B \in \mathcal{A}$,

$$A \prec_{SP} B \Leftrightarrow DR(A) \triangleleft_{SP} DR(B). \quad (4.5)$$

The associated non-strict argument preference is

$$A \preceq_{SP} B \Leftrightarrow [DR(A) \triangleleft_{SP} DR(B) \text{ or } DR(A) = DR(B)], \quad (4.6)$$

These equations are the same as Equations 3.8 and 3.9 respectively.³²

Example 13. (Example 8 continued) Suppose we define \prec_{SP} by Equation 4.5 with this new $<_{SP}$. We have both $A \not\prec_{SP} B$ and $B \not\prec_{SP} A$. This means there are two stable extensions: \mathcal{E}_1 which contains A_0 and A , and \mathcal{E}_2 which contains A_0 and B . The conclusion set of these stable extensions correspond respectively to E_1 and E_2 .

So given a PDT T where the default priority $<$ is not necessarily total, we construct the set of arguments \mathcal{A} and define the attack relation \rightarrow as in Section 4. We define the non-strict argument preference relation \lesssim_{SP} as in Equations 4.6, 4.5, 4.4 and 4.3, given $< \cong <_D$. The *attack graph of the PDT T* is the structure $AG(T) := \langle \mathcal{A}, \rightarrow, \lesssim_{SP} \rangle$. The *defeat graph of the PDT (T)* is the structure $DG(T) := \langle \mathcal{A}, \hookrightarrow \rangle$, where \hookrightarrow is defined as in Equation 2.3 under the argument preference relation \lesssim_{SP} .

4.2 The Representation Theorem for Partial Order Default Priorities

We now generalise Theorem 3.14 to the case where $<_D$ is a partial order. Our proof strategy is to leverage as much of Theorem 3.14 as possible. The difference here is that our default priority $< \cong <_D$ is now partial. In the previous section, we saw how the linearisation square (Theorem 4.4) related the lift $<_D \mapsto <_{SP}$ to the lift $<_D^+ \mapsto <_{SP}^+$ where $<_D^+$ is a linearisation of $<_D$. We now apply the linearisation square to relate partial order $<_D$ with their linearisations $<_D^+$ in the case of the defeat graphs generated and their stable extensions. Specifically, if $<_D^+$ is a linearisation of $<_D$, then the defeat graph of the former is a spanning subgraph of the latter. Further, the unique stable extension in the former case is still a well-defined stable extension in the latter case. The next two sections establish these results, which will then be used to prove the generalised representation theorem.

4.2.1 Linearisation of the Argument Preference Relation and Spanning Subgraphs

Recall from graph theory that $G' := \langle V, E' \rangle$ is a *spanning subgraph* of $G := \langle V, E \rangle$ iff $E' \subseteq E$, and we write $G' \subseteq_{\text{span}} G$. For spanning argument subframeworks, stable extensions are preserved as long as you do not add conflicts between arguments in the stable extension.

Lemma 4.6. *Let $AF := \langle A, \rightarrow \rangle$ be an abstract argumentation framework. Let $AF' := \langle A, \rightarrow' \rangle$ be a spanning subgraph of AF . If \mathcal{E} is a stable extension of AF' and $\mathcal{E}^2 \cap \rightarrow = \emptyset$, then \mathcal{E} is also a stable extension of AF .*

³²It can be shown that \lesssim_{SP} in the partial order case is *not* transitive, unlike in the total case (Lemma 3.5, also recall Footnote 15), but is acyclic. We will discuss this in future work.

Proof. By assumption, \mathcal{E} is cf because it is a stable extension. Let $b \notin \mathcal{E}$, then $\mathcal{E} \rightarrow' b$, but as $\rightarrow' \subseteq \rightarrow$ by definition, we also have $\mathcal{E} \rightarrow b$. Therefore, \mathcal{E} is a stable extension of AF . \square

Linearising the structure preference order $<_{SP}$ on the rules also linearises the set comparison relation \triangleleft_{SP} and the argument preference \prec_{SP} by Lemma 3.4.

Lemma 4.7. *Let $<_{SP}$ be the output of Equation 4.3 for some input $<_D$. Let $<_{SP}^+$ be a linearisation of $<_{SP}$. Let the binary relations on $\mathcal{P}_{fin}(\mathcal{R}_d)$, \triangleleft_{SP} and \triangleleft_{SP}^+ , be obtained by applying Equation 4.4 to $<_{SP}$ and $<_{SP}^+$ respectively. Then*

1. $\triangleleft_{SP} \subseteq \triangleleft_{SP}^+$,
2. $\prec_{SP} \subseteq \prec_{SP}^+$, where \prec_{SP} is the strict part of Equation 4.5 on \triangleleft_{SP} and analogously for \prec_{SP}^+ on \triangleleft_{SP}^+ , and
3. $\prec_{SP} \subseteq \prec_{SP}^+$.

Proof. (1) Let $\Gamma, \Gamma' \in \mathcal{P}_{fin}(\mathcal{R}_d)$ be arbitrary. Assume $\Gamma \triangleleft_{SP} \Gamma'$. Then by Equation 4.4, this is equivalent to $(\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x <_{SP} y$, which by our assumption implies that $(\exists x \in \Gamma - \Gamma') (\forall y \in \Gamma' - \Gamma) x <_{SP}^+ y$, and hence $\Gamma \triangleleft_{SP}^+ \Gamma'$. (2) Let $A, B \in \mathcal{A}$ be arbitrary. We have $A \prec_{SP} B \Leftrightarrow DR(A) \triangleleft_{SP} DR(B)$. From the first result, $DR(A) \triangleleft_{SP} DR(B)$ then $DR(A) \triangleleft_{SP}^+ DR(B)$. (3) This follows trivially from Equation 4.5. \square

We now prove the converse of Lemma 4.7.

Lemma 4.8. *Let T be a PDT, $< \cong <_D$ and $<_{SP} = F(<_D)$. Let $<_{SP}^+$ be a linearisation of $<_{SP}$. Let \prec_{SP} and \prec_{SP}^+ be the lift of $<_{SP}$ and $<_{SP}^+$ respectively from \mathcal{R}_d to an argument preference relation on \mathcal{A} in the usual way (Equations 4.4 and 4.5). Let $DG(T) := \langle \mathcal{A}, \hookrightarrow \rangle$ and $DG^+(T) := \langle \mathcal{A}, \hookrightarrow^+ \rangle$ be the respective defeat graphs of the attack graphs $AG(T) := \langle \mathcal{A}, \rhd, \prec_{SP} \rangle$ and $AG^+(T) := \langle \mathcal{A}, \rhd, \prec_{SP}^+ \rangle$. Then $DG^+(T) \subseteq_{span} DG(T)$.*

Proof. Clearly both $DG(T)$ and $DG^+(T)$ have the same vertex set \mathcal{A} . We show that $\hookrightarrow^+ \subseteq \hookrightarrow$. Let $A, B \in \mathcal{A}$ be arbitrary such that $A \hookrightarrow^+ B$. Suppose $B' \subseteq_{arg} B$ is the argument defeated by A at its top rule. By Equation 2.3, $A \rhd B'$ and $A \not\prec_{SP}^+ B'$. It is sufficient to show that $A \not\prec_{SP} B'$. By Lemma 4.7, we have $\prec_{SP} \subseteq \prec_{SP}^+$ meaning that if $A \not\prec_{SP}^+ B'$ then $A \not\prec_{SP} B'$. Hence, $A \hookrightarrow B'$ and so $A \hookrightarrow B$. It follows that $\hookrightarrow^+ \subseteq \hookrightarrow$. \square

4.2.2 Existence of Stable Extensions in the Partial Order Case

For a PDT T , its defeat graph $DG(T)$ has stable extensions that do not have to be unique.

Theorem 4.9. *Let T be a PDT, with attack graph $AG(T)$ and defeat graph $DG(T)$ where, as usual, $< \cong <_D$, $<_{SP} = F(<_D)$ by Equation 4.3, and \prec_{SP} is defined from $<_{SP}$ using Equations 4.4 and 4.5. The defeat graph $DG(T)$ has a stable extension that is not in general unique.*

Proof. The PDT T has some extension E , which is the unique stable extension of an LPDT $T^+ := \langle D, W, <^+ \rangle$, where $<^+$ is the linearisation of $<$ that generates E . Given that $< \cong <_D$ and $<^+ \cong <_D^+$, we know that $<_D^+$ is also a linearisation of $<_D$. By the linearisation square (Theorem 4.4), $F(<_D^+) =: <_{SP}^+$ is a linearisation of $F(<_D) =: <_{SP}$.

The LPDT T^+ has an attack graph $AG(T^+) := \langle \mathcal{A}, \rightarrow, \lesssim_{SP}^+ \rangle$, where \lesssim_{SP}^+ is calculated from $<_{SP}^+$. As $<_{SP}^+$ is a linearisation of $<_{SP}$, the defeat graph of T^+ , $DG(T^+) := \langle \mathcal{A}, \hookrightarrow^+ \rangle$ is a spanning subgraph of $DG(T)$ by Lemma 4.8.

The LPDT T^+ has E as its unique PDE. By the representation theorem for LPDTs (Theorem 3.14), there exists a unique stable extension \mathcal{E} of $DG(T^+)$ such that $Conc(\mathcal{E}) = E$. By Corollary 3.21, $\mathcal{E}^2 \cap \rightarrow = \emptyset$. $DG(T)$ differs from $DG(T^+)$ by their argument preference relations, as the attack relation is the same for both. As no attacks are introduced to \mathcal{E} , \mathcal{E} is also cf in $DG(T)$ by Equation 2.3. Therefore, by Lemma 4.6, \mathcal{E} is also a stable extension of $DG(T)$. Therefore, $DG(T)$ also has \mathcal{E} as a stable extension.

To show that this stable extension is not in general unique, consider the PDT $\langle \{d_1 := \frac{a}{a}, d_2 := \frac{\neg a}{\neg a}\}, \emptyset, \emptyset \rangle$. We can construct the arguments $A := [\Rightarrow a]$ and $B := [\Rightarrow \neg a]$, which symmetrically attack each other on their conclusions. As $< = \emptyset$, we have $<_D = <_{SP} = \emptyset$ from Equation 4.3. Therefore, $A \not\prec_{SP} B$ and $B \not\prec_{SP} A$ and we have two stable extensions: one where A is justified (and B is not justified), and the other where B is justified (and A is not justified). \square

4.2.3 Proof of the Representation Theorem

We prove the representation theorem in this section. Our technique is to relate a partial order on the defaults $<$ with one of its possible linearisations $<^+$, and invoke the first representation theorem (Theorem 3.14) for this linearisation. We know given an extension E of some PDT T there is a linearisation $<^+$ of $<$ generating E . We now establish an analogous result on the side of argumentation: for every stable extension \mathcal{E} of $DG(T)$, there is a linearisation $<_{SP}^+$ of $<_{SP}$ on \mathcal{R}_d such that $<_{SP}^+$ constructs \mathcal{E} via Algorithm 1.

We show $<_{SP}^+$ exists given $<_{SP}$ by construction, which will make use of a *partial linearisation* of an order $<$ on P . Let $\langle P, < \rangle$ be a poset and $U \subseteq P$. Let $<_U := < \cap U^2$ be the partial order $<$ restricted to U . Let $<_U^+$ be a linearisation of $<_U$ on U . We define

$$<_U^{\text{par}} := TrCl(<_U^+ \cup <), \quad (4.7)$$

where $TrCl$ denotes the transitive closure. It can be shown that given $\langle P, < \rangle$ and U , $<_U^{\text{par}}$ is a strict partial order on P extending $<$, which is linear when restricted to the set U . Further, $<_U^{\text{par}}$ is not unique because there could be many possible linearisations of $<$ over U .

Lemma 4.10. *Let T be a PDT with defeat graph $DG(T)$ where $<_{SP}$ lifts to \lesssim_{SP} . Let \mathcal{E} be a stable extension of $DG(T)$. There exists a linearisation $<_{SP}^+$ of $<_{SP}$ such that \mathcal{E} is the output of Algorithm 1 with $<_{SP}^+$ as input.*

Proof. Given \mathcal{E} , let $R^+ := DR(\mathcal{E})$ and $R^- = \mathcal{R}_d - R^+$. Define $<_0 := <_{SP, R^-}^{\text{par}}$, which is Equation 4.7 with $\langle P, < \rangle = \langle \mathcal{R}_d, <_{SP} \rangle$ and $U = R^-$. As \mathcal{R}_d is finite, WLOG let $R^- := \{s_1, s_2, \dots, s_m\}$ for some $m \in \mathbb{N}$, such that $i < j \Leftrightarrow s_j <_0 s_i$. so s_1 is $<_0$ -greatest on R^- . For $1 \leq i \leq m$, define the set

$$\text{nonlower}_{<_{i-1}}(s_i) := U_i := \{r \in \mathcal{R}_d \mid r \not<_{i-1} s_i\}. \quad (4.8)$$

For $0 \leq i \leq m-1$ we extend $<_i$ to a new partial order $<_{i+1} := <_{i, U_{i+1}}^{\text{par}}$, which is Equation 4.7 with $\langle P, < \rangle = \langle \mathcal{R}_d, <_i \rangle$ and $U = U_i$ (Equation 4.8), such that s_i is the $<_i$ -least element on the set U_i . Once we reach $<_m$ we take a final linearisation of $<_m$ to get $<_{SP}^+$.³³ This construction therefore gives an increasing sequence of partial orders $<_{SP} \subseteq <_0 \subseteq <_1 \subseteq \dots \subseteq <_m \subseteq <_{SP}^+$ on \mathcal{R}_d . Clearly, $<_{SP}^+$ is a well-defined linearisation of $<_{SP}$ by construction.

To show that $<_{SP}^+$ generates \mathcal{E} when input into Algorithm 1, consider $s_i \in R^-$. We assume no defeasible rule is unnecessary, i.e. $(\forall r \in \mathcal{R}_d) (\exists A \in \mathcal{A}) r \in DR(A)$. In other words, each defeasible rule is used in some argument.³⁴ Therefore, there is some argument $B_i \in \mathcal{A}$ such that $\text{TopRule}(B_i) = s_i$. By how $<_{SP}$ is defined, s_i is the $<_{SP}$ -least rule in $DR(B_i)$. By Lemma 3.12, $B_i \notin \mathcal{E}$ so $\mathcal{E} \hookrightarrow B_i$. Let $A_i \hookrightarrow B_i$ for $A_i \in \mathcal{E}$. This would mean $A_i \not<_{SP} B_i$.

We show that for the set of defeasible rules U_i associated with rule s_i as defined in Equation 4.8, $(\forall r \in DR(A_i)) r \in U_i$. Assume for contradiction that $(\exists r \in DR(A_i)) r \notin U_i$, then there is some $r_0 \in DR(A_i)$, $r_0 <_{i-1} s_i$. By the properties of Equation 4.7, we can show that $r_0 <_{i-1} s_i <_{i-1} s_{i-1} <_{i-1} \dots <_{i-1} s_1$. Therefore, r_0 cannot be in the sets U_j for any $j < i$, and so could not have been linearised above s_j for $j < i$ in any of the previous stages. Therefore, $r_0 <_{SP} s_i$. As s_i is $<_{SP}$ -least in $DR(B_i)$ by being the top rule of B_i , $r_0 \notin DR(B_i)$ and hence there is some rule, $r_0 \in DR(A_i) - DR(B_i)$, such that for all rules $x \in DR(B_i) - DR(A_i)$, $r_0 <_{SP} x$. Therefore, $A_i \prec_{SP} B_i$ – contradiction, as $A_i \hookrightarrow B_i$. Therefore, all the defeasible rules of A_i are in U_i , and in the linearisation process where $<_{i-1}$ is linearised over U_i into $<_i$ such that s_i is $<_i$ -minimal in U_i , we have ensured that at least one defeater of B_i will be constructed by Algorithm 1 and included in \mathcal{E} prior to the consideration of the rule s_i . As i is arbitrary, we have shown that the final linearisation $<_{SP}^+$ ensures that all arguments containing rules in R^- are defeated and excluded from \mathcal{E} . Therefore, Algorithm 1, upon input from $<_{SP}^+$, generates exactly \mathcal{E} . \square

We give two concrete examples of the construction of $<_{SP}$ in Lemma 4.10.

Example 14. Consider the PDT $\langle \{d_1 := \frac{a}{a}, d_2 := \frac{\neg a}{\neg a}\}, \emptyset, \emptyset \rangle$ from the proof of Theorem 4.9. Translating to argumentation, there are two arguments $A := [\Rightarrow a]$ and $B := [\Rightarrow \neg a]$ which attack each other at their conclusions. Clearly, $<_{SP} = \emptyset$ and there are two stable extensions: \mathcal{E}_1 such that $A \in \mathcal{E}_1$ and \mathcal{E}_2 such that $B \in \mathcal{E}_2$. Suppose we choose the stable extension \mathcal{E}_1 and construct a linearisation of \emptyset that generates \mathcal{E}_1 . We have $R^+ = \{r_1\}$ and $R^- = \{r_2\}$. Vacuously, $<_{SP}$

³³Simple examples can be devised where $<_m$ is not a total order on \mathcal{R}_d .

³⁴This is a fair assumption to make given that PDTs typically do not have defaults that are excluded from all extensions.

is already linear on $\{r_2\}$ so $<_0 = <_{SP}$. We then consider $nonlower_{<_{SP}}(r_2) = \{r_1, r_2\}$. We linearise $<_{SP}$ such that r_2 is smaller than all other elements in $nonlower_{<_{SP}}(r_2)$, so $r_2 <_{SP}^+ r_1$. This is indeed the linearisation of $<_{SP}$ that generates \mathcal{E}_1 .

Example 15. Let $\mathcal{K}_n = \emptyset$ and $r_1 := (\Rightarrow \neg b)$, $r_2 := (\Rightarrow a)$, $r_3 := (a \Rightarrow b)$, $r_4 := (\Rightarrow c)$, $r_5 := (c \Rightarrow \neg b)$, $r_6 := (\Rightarrow b)$. Define $A := [\Rightarrow \neg b]$, $B := [[\Rightarrow a] \Rightarrow b]$, $C := [[\Rightarrow c] \Rightarrow \neg b]$ and $D := [\Rightarrow b]$. We illustrate these arguments in Figure 4.2.

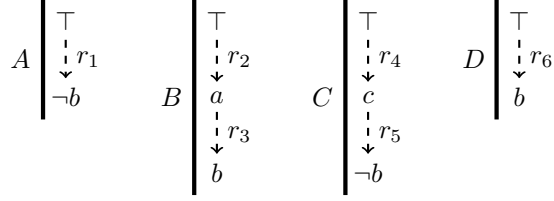


Figure 4.2: The arguments of Example 15.

Suppose $<_{SP}$ is such that $r_6 <_{SP} r_5 <_{SP} r_4$, $r_5 <_{SP} r_3 <_{SP} r_1$ and $r_3 <_{SP} r_2$. It can be shown that $D \prec_{SP} C \prec_{SP} B \prec_{SP} A$ hence $A \hookrightarrow B \hookrightarrow C \hookrightarrow D$ (notice $A \hookrightarrow D$ as well). The stable extension therefore contains A , C and $[\Rightarrow a]$, so $R^- = \{r_3, r_6\}$. As $r_6 <_{SP} r_3$, $<_{SP}$ is already linear on R^- , so $<_0 = <_{SP}$. Now consider $nonlower_{<_0}(r_3) = \{r_1, r_2, r_3, r_4\}$ and linearise $<_{SP}$ over this set such that r_3 is the smallest element in $nonlower_{<_0}(r_3)$, so suppose $<_1$ is $r_3 <_1 r_1 <_1 r_2 <_1 r_4$. Now consider r_6 , but $nonlower_{<_1}(r_6) = \mathcal{R}_d$ and is already linear, so we take $<_{SP}^+$ to be the chain 653124 when written in abbreviated form (see Footnote 31, page 37). This $<_{SP}^+$, when input into Algorithm 1, will generate \mathcal{E} .

We now apply Lemma 4.10 to prove a more general representation theorem.

Theorem 4.11. (*The Representation Theorem for Partial Order Default Priorities*) Let $AG(T)$ be the attack graph corresponding to a PDT T , where the default priority $<$ is not necessarily total, with defeat graph $DG(T)$ under \preceq_{SP} as defined by Equations 4.3, 4.4 and 4.5.

1. Let E be an extension of T . Then there exists a corresponding stable extension $\mathcal{E} \subseteq \mathcal{A}$ of $DG(T)$ such that $Conc(\mathcal{E}) = E$.
2. Let $\mathcal{E} \subseteq \mathcal{A}$ be a stable extension of $DG(T)$, then $Conc(\mathcal{E})$ is an extension of T .

Proof. Proof of part 1: Let E be an extension of T , then there exists a LPDT $T^+ := \langle D, W, <^+ \rangle$ where $<^+$ is a linerisation of $<$ that generates the extension E (Equation 2.5). Consider the defeat graph $DG(T^+)$. By Theorem 3.14, there exists a stable extension \mathcal{E} of $DG(T^+)$ such that $Conc(\mathcal{E}) = E$. Arguing as

in the proof of Theorem 4.9 where $DG(T^+) \subseteq_{\text{span}} DG(T)$, \mathcal{E} is also a stable extension of $DG(T)$, and it satisfies $\text{Conc}(\mathcal{E}) = E$.

Proof of part 2: Let \mathcal{E} be a stable extension, which exists by Theorem 4.9. By Lemma 4.10, there is a linearisation $<_{SP}^+$ of $<_{SP}$ such that Algorithm 1 returns \mathcal{E} upon input $<_{SP}^+$. Consider the LPDT $T_1^+ := \langle D, W, <_1^+ \rangle$, where $<_1^+ \cong <_{SP}^+$. By Section 3, this has a defeat graph $DG(T_1^+)$ with unique stable extension \mathcal{E} . The set $\text{Conc}(\mathcal{E})$ is an extension of T_1^+ by Theorem 3.14. Clearly $\text{Conc}(\mathcal{E})$ is also an extension of $\langle D, W, <_1 \rangle$, where $<_1 \cong <_{SP}$. By Lemma 4.5, $\text{Conc}(\mathcal{E})$ is also an extension of $\langle D, W, < \rangle = T$. \square

Under the generalised SP argument preference \lesssim_{SP} , this representation theorem means that PDL is also sound and complete with respect to its argumentation semantics in the case where the default priority $<$ is not necessarily total.

4.3 Satisfaction of Rationality Postulates

In this section we will state and prove a version of Theorem 3.20, which is that the rationality postulates [10] hold for the stable extensions of the defeat graph, instead for all complete extensions. We will discuss the possibility for a general proof in Section 5.

Theorem 4.12. (*Rationality Theorem for Stable Extensions*) *Let $T := \langle D, W, < \rangle$ be a PDT. Let its corresponding attack graph be $AG(T) := \langle \mathcal{A}, \rightarrow, \lesssim_{SP} \rangle$ where $< \cong <_D \mapsto <_{SP}$ by Equation 4.3, and \lesssim_{SP} is defined in terms of $<_{SP}$ using Equations 4.4 and 4.5. Let $DG(T) := \langle \mathcal{A}, \hookrightarrow \rangle$ be the corresponding defeat graph. All stable extensions of $DG(T)$ satisfy the Caminada-Amgoud rationality postulates.*

Proof. Given T , let \mathcal{E} be any stable extension of $DG(T)$.

1. To show that \mathcal{E} is subargument closed, let $A \in \mathcal{E}$ and let $B \subseteq_{\text{arg}} A$. Assume for contradiction that $B \notin \mathcal{E}$, then $\mathcal{E} \hookrightarrow B$ and hence there is some $C \in \mathcal{E}$ such that $C \hookrightarrow B$. Therefore, $C \hookrightarrow A$. This means \mathcal{E} is not cf – contradiction. Therefore, $B \in \mathcal{E}$ as well, and \mathcal{E} is thus subargument closed.
2. Theorem 4.11 states that $\text{Conc}(\mathcal{E})$ is an extension of T , which is deductively closed. Therefore, $\text{Conc}(\mathcal{E})$ is closed under strict rules.
3. As W is consistent and $\text{Conc}(\mathcal{E})$ is an extension of T , $\text{Conc}(\mathcal{E})$ must also be consistent and its deductive closure is consistent.

This shows the result. \square

In conclusion, all stable extensions are normatively rational. This generalises the rationality theorem (Theorem 3.20) to the case where $<_D$ is partial, although only for stable extensions.

4.4 Summary

In this section, we have generalised our sound and complete instantiation of ASPIC⁺ to PDL to the case where the default priority is not necessarily a total order. The main challenge is generalising the SP argument preference \lesssim_{SP} from a total default priority to a partial default priority. We devise a sorting $F(<_D) = <_{SP}$ such that $<_{SP}$ sorts $<_D$ in a way that respects the argument structure, the defeasible rule preference $<_D$, and the incomparability of rules (Section 4.1). This preference has the correct properties to preserve the correspondence between the inferences of the underlying PDT and the conclusions of justified arguments (Section 4.2, Theorem 4.11). We have also shown that each stable extension satisfies the rationality postulates (Section 4.3, Theorem 4.12).

5 Conclusions

We have endowed Brewka’s PDL [7] with argumentation semantics using ASPIC⁺ [16]. This is achieved by representing PDL in ASPIC⁺ (Sections 3.1 and 4), discussing which argument preference relations can be suitable for the correspondence of inferences (Sections 3.2 and 4.1), proving that the inferences do correspond under an appropriate preference relation \lesssim_{SP} (Sections 3.3 and 4.2), and that this instantiation is rational (Sections 3.4 and 4.3). As explained in Section 1, this allows us to interpret the inferences of PDL as conclusions of justified arguments, clarifying the reasons for accepting or rejecting a conclusion. The argumentative characterisation of PDL provides for distributed reasoning in the course of deliberation and persuasion dialogues. This would allow BOID agents with PDL representations of mental attitudes to exchange arguments and counterarguments when deliberating about which goals to select, and thus which actions to pursue [9].

5.1 Related Work

As mentioned in Section 1, there are many existing argumentative characterisations of non-monotonic logics (e.g. [11, 12]). However, there has been relatively little work in using defeasible rules to represent defaults, because the defeasible components of arguments are often captured in the premises [5, 16]. Reiter’s default logic (DL) [19], as a partial special case³⁵ of Brewka’s PDL [7, Proposition 6], has been endowed with sound and complete argumentation semantics by Dung [11, Section 4.1]. However, DL cannot handle priorities and as a result draws counter-intuitive inferences. We know that conflicts between defaults often occur and priorities are an intuitive and high-level way of resolving such conflicts [7, 8]. It is therefore important to investigate how preferences can also be incorporated into any argumentation semantics. ASPIC⁺ is a good framework to achieve this because it is designed to handle preferences.

³⁵i.e. in the case where all defaults are normal defaults.

5.2 Future Work

Brewka’s preferred subtheories (PS) [6] has been endowed with argumentation semantics by Modgil and Prakken using ASPIC⁺ [16, Section 5.3.2]. Given that PS is a special case of PDL [6], it is interesting to see how the argumentation semantics are related. It can be shown that instantiating the argumentation semantics of PDL in this paper to the case of supernormal defaults and empty facts will recover an argumentation semantics isomorphic to the argumentation semantics of Modgil and Prakken. However, whereas Modgil and Prakken assume that arguments must be consistent, the results of Section 3.4.3 lifts this assumption when we specialise our argumentation semantics. We will articulate this in future work.

It will also be interesting to see how an argumentation semantics for Reiter’s normal DL [19, Section 3] can be recovered by setting $<_D = \emptyset$ [7, Proposition 6], and comparing this to Dung’s argumentation semantics for DL. However, Dung’s argumentation semantics also accommodates non-normal defaults. How would ASPIC⁺ incorporate non-normal defaults? At first glance it should involve the naming function and undercuts (Section 2.2), but how can soundness and completeness be proven? How can the argumentation semantics help us understand the interaction of explicit default priority relations with the implicit priority of non-normal defaults [21]? Future work will explore further properties of this argumentation semantics.

ASPIC⁺ can be used to generalise PDL. For example, we know that extensions do not have to exist for non-normal default logic, which corresponds to the failure for stable extensions to exist in the argumentation semantics [11, Section 4.1]. We can then consider the justified arguments under different Dung semantics, but what would these other notions of justified arguments mean for PDL?

Another reason for considering different Dung semantics is to show whether the rationality postulates holds for complete extensions in general. So far we have shown a special case of rationality for the stable extensions only (Section 4.3). What would the complete extensions look like in this case? How are they related to the other Dung semantics [11, Section 2.3]? Alternatively, one can invoke the theory of ASPIC⁺, which states that normative rationality automatically follows if the instantiation is *well-defined* with a *reasonable* argument preference relation [16, Definitions 12 and 18]. Although it is easy to see that our instantiation is well-defined if the underlying PDT is consistent, it is not obvious whether \lesssim_{SP} in the partial order case is reasonable. This will be the subject of future work.

Finally, we have argued that endowing PDL with argumentation semantics provides for distributed reasoning amongst agents (in particular BOID agents for which PDL has been used to generate *individual* agents’ goals). Such distributed reasoning in the form of dialogue can be formalised as a generalisation of argument game proof theories for Dung frameworks [15], whereby agents not only can submit arguments, but locutions that implicitly define arguments providing the reasons for a given claim. We will investigate this in future work.

References

- [1] K. Atkinson, T. Bench-Capon, and P. McBurney. A Dialogue Game Protocol for Multi-Agent Argument over Proposals for Action. *Autonomous Agents and Multi-Agent Systems*, 11(2):153–171, 2005.
- [2] S. Barberà, W. Bossert, and P. K. Pattanaik. Ranking Sets of Objects. In S. Barberà, P. J. Hammond, and C. Seidl, editors, *Handbook of Utility Theory, Volume 2 – Extensions*, chapter 17, pages pp. 893 – 978. Springer Science + Business Media, 1st edition, 2004.
- [3] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni. Introduction to Structured Argumentation. *Argument & Computation*, 5(1):1–4, 2014.
- [4] A. Bochman. *Handbook of the History of Logic*, volume 8, chapter Nonmonotonic Reasoning, pages 557–632. Elsevier, 2007.
- [5] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63–101, 1997.
- [6] G. Brewka. Preferred Subtheories: An Extended Logical Framework for Default Reasoning. In *IJCAI*, volume 89, pages 1043–1048, 1989.
- [7] G. Brewka. Adding Priorities and Specificity to Default Logic. In *Logics in Artificial Intelligence*, pages 247–260. Springer, 1994.
- [8] G. Brewka and T. Eiter. Prioritizing Default Logic. In *Intellectics and Computational Logic*, pages 27–45. Springer, 2000.
- [9] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal Generation in the BOID Architecture. *Cognitive Science Quarterly Journal*, 2(3-4):428–447, 2002.
- [10] M. Caminada and L. Amgoud. On the Evaluation of Argumentation Formalisms. *Artificial Intelligence*, 171(5):286–310, 2007.
- [11] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n -Person Games. *Artificial Intelligence*, 77:321–357, 1995.
- [12] G. Governatori, M. J. Maher, G. Antoniou, and D. Billington. Argumentation Semantics for Defeasible Logic. *Journal of Logic and Computation*, 14(5):675–702, 2004.
- [13] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.

- [14] S. Modgil. An Argumentation Based Semantics for Agent Reasoning. In M. Dastani, A. El Fallah Seghrouchni, J. Leite, and P. Torroni, editors, *Languages, Methodologies and Development Tools for Multi-Agent Systems*, volume 5118 of *Lecture Notes in Computer Science*, pages 37–53. Springer Berlin Heidelberg, 2008.
- [15] S. Modgil and M. Caminada. Proof Theories and Algorithms for Abstract Argumentation Frameworks. In *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.
- [16] S. Modgil and H. Prakken. A General Account of Argumentation with Preferences. *Artificial Intelligence*, 195:361–397, February 2013.
- [17] S. Modgil, F. Toni, et al. The Added Value of Argumentation. In S. Ossowski, editor, *Agreement Technologies*, volume 8, pages 357–403. Springer Netherlands, 2013.
- [18] H. Prakken. An Abstract Framework for Argumentation with Structured Arguments. *Argument and Computation*, 1(2):93–124, 2010.
- [19] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81–132, 1980.
- [20] R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 2:147–186, 1987.
- [21] R. Reiter and G. Criscuolo. On Interacting Defaults. In *IJCAI*, volume 81, pages 270–276, 1981.
- [22] R. Thomason. “Logic and Artificial Intelligence”, The Stanford Encyclopedia of Philosophy (Winter 2013 Edition), Edward N. Zalta (ed.), forthcoming URL = <http://plato.stanford.edu/archives/win2013/entries/logic-ai/>.
- [23] A. P. Young, S. Modgil, and O. Rodrigues. Argumentation Semantics for Prioritised Default Logic. *arXiv preprint arXiv:1506.08813*, 2015.
- [24] A. P. Young, S. Modgil, and O. Rodrigues. Prioritised Default Logic as Rational Argumentation. In J. Thangarajah, K. Tuyls, C. Jonker, and S. Marsella, editors, *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, 2016.